UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI MATEMATICA
Corso di Laurea Magistrale in Matematica

Radial Basis Function Network for ODEs: application to diabetes and insulin therapy models

Laureanda: Giulia Antinori

Relatore: Prof. Stefano De Marchi

Correlatore: Prof. Wolfgang zu Castell

anno accademico 2011/2012

Ai miei genitori.

Ringraziamenti

Desidero innanzitutto ringraziare il mio relatore Prof. Stefano De Marchi non solo per le numerose ore che ha dedicato alla mia tesi, ma anche per avermi sempre aiutato con tutte le controversie burocratiche e per avermi permesso di vivere una bellissima esperienza come quella dell'Erasmus.

Ringrazio anche il mio correlatore Prof. Wolfgang zu Castell, non solo per il supporto scientifico alla tesi ma anche per avermi accolto a Monaco e all'Helmholtz Zentrum.

La mia gratitudine va anche al Prof. Rick Beatson per i preziosi consigli durante la ricerca.

Non posso non ringraziare Martina, Michael, Phaedra, Meltem e Martin per aver sempre risposto alle mie domande e anche per le pause caffé .

Infine vorrei ringraziare tutte le persone che mi hanno sempre sostenuto nel corso di questi anni, la mia famiglia, il mio fidanzato Mattia, tutti i miei compagni di corso con i quali ho condiviso tante ore di studio ma anche di divertimento e tutte le mie compagne di collegio.

Acknowledgements

First and foremost I would like to thank my supervisor Prof. Stefano De Marchi, not only for all the hours he dedicated to my thesis, but also for always helping me with all the bureaucratic disputes and letting me live a beautiful experience as that of Erasmus exchange programme.

I am also very grateful to Prof. Wolfgang zu Castell, not only for the scientific support to the thesis but also for welcoming me to Munich and to the Helmholtz Zentrum.

My gratitude goes also to Prof. Rick Beatson for his valuable advices during the research.

I can not forget to thank Martina, Michael, Phaedra, Meltem and Martin for always answering my questions and for the coffee breaks.

Finally I would like to thank all the people who have always supported me during these years, my family, my boyfriend Mattia, all my classmates, with whom I shared many hours of study but also of fun, and all my friends at the university residence.

Introduction

Mathematics and biology have a synergistic relationship. The linkage between these two sciences is embodied in the reciprocal contributions they make to each other: biology generates complex problems and mathematics can provide ways to understand them. Indeed, mathematics provides a simple representation of the reality through the creation of models and their solution by the extensive use of numerical and symbolic computational software.

In many cases, understanding quantitatively a natural process, leads to a differential equation model.

These considerations led our attention to the study of a new method for the numerical solution of ODEs systems which arise from biological models. In particular, we consider a method which makes use of a Radial Basis Function Network (RBFN) and it is applied to a biological model for diabetes and insulin therapy.

A RBFN, as the name says, is the union of two powerful tools: Radial Basis Functions (RBF) and Neural Networks. Both these tools have gained much attention in recent years thanks to the fact that they can be applied to many different areas of science and engineering.

For what concerns RBF, they were used for the first time in the solution of the real multivariate interpolation problem (cf. [2]) and their great success is mainly due to the fact that they are a meshless method, i.e. they are not related to a grid (see Chapter 1).

With the term "Neural Network" we improperly indicate Artificial Neural Network (ANN), i.e. a net of artificial neurons that can be used both to gain an understanding of biological neural networks and to solve artificial intelligence problems without necessarily creating a model of a real biological system. The architecture of an ANN is inspired by the brain structure. Indeed, an ANN is a group of interconnected units, called *neurons*, which are related by linkages, called *synapses*, and organized in layers. The data processing takes place in the neurons. From a mathematical point of view this means that the input data are passed to a function, typically non-linear, called *activation function*.

In this thesis we focus on the specific case of a RBFN, that is an artificial neural network whose activation function is a RBF (see Chapter 2). This choice is motivated by the fact that a RBFN is a "universal approximator", in the sense that almost every function can be approximated by an artificial neural network (cf. [11, 12]). Moreover, differently from a common approach which used a backward propagation algorithm to teach the artificial network how to perform a task, in the case of a RBF network the learning phase consists in the solution of an interpolation problem. Thus we do not need to investigate the convergence problems of the backward propagation algorithm and in the meanwhile we also reduce the computational cost of the network training phase.

Once we have an explicit expression for the RBF network, we proceed in three steps:

1. study of a method for the simultaneous approximation of a function and its derivatives (see Chapter 3);

2. use of the previous result to solve an ordinary differential equation of arbitrary order (see Chapter 4);

3. application of the RBF theory to the vectorial valued case and use of this theory, together with the result in 2, to solve an ODEs system (see Chapter 5).

For what concerns the first step, we refer to [6]. The key idea in that paper is that if we approximate the function with the RBFN, then, simply differentiating the network, we may obtain an approximation also for the derivatives. This leads to what they call a Direct Radial Basis Function Network method (DRBFN). On the other hand, if we approximate the nth derivative with the RBFN and then we integrate it, we obtain the Indirect Radial Basis Function Network method (IRBFN).
We do not limit to the numerical experiments and we make a deep analysis on the network functioning. It will be clear from these numerical results that some RBF provide better accuracy than others. This is the reason why we have repeated the simulations with different kernels.
A first contribution in this thesis is the observation that this different accuracy can be justified mainly by two reasons. The first concerns the kernel we choose for the approximation: those kernels φ such that $\varphi(r) \to \infty$ as $r \to \infty$ have better localization properties compared with those ones that tend to zero as $r \to \infty$. It is worth noting that localization properties are highly important to the success of general approximation methods. The second reason is related to the function f to be approximated. It is well known that a good approximation is guaranteed only if f belongs to the native space of the kernel φ. We did also an error analysis using error estimates in [7], for different kind of functions and kernels.
We also state an existence theorem for the RBFN, under some assumptions on the radial function φ, the proof is in [12].
For numerical experiments of the second step above we refer to [9]. Moreover, since in the numerical construction of RBFN or in the solution of the ordinary differential equation, we face with the solution of an ill conditioned linear system, an analysis of the matrix condition number has been necessary. Thus, we illustrate two methods to reduce the conditioning: the Riley method [16] and a domain scaling. The results suggest that both these approaches depend on the specific example we take into account. However the scaling performs much better.
Finally, in Chapter 5, we extend the RBF approach to the vectorial case. All the theoretical results of Chapter 1 are still valid if instead of considering one single kernel, we consider a matrix of kernels. In particular the interpolation problem, which arises from the RBFN definition, is well posed and admits unique solution if we multiply the matrix kernels by a positive definite matrix, say C. How the coefficients of C should be chosen

is not automatically determined. Thus, a parameter analysis has been done, see Section 5.2.
The last Chapter can be seen as an original generalization of the previous. Indeed the RBFN permits the representation of the ODEs system in a compact form, which reflects the matrix form of the interpolation problem in the vectorial case.

Contents

1 Radial Basis Functions: essential theory and properties **7**
- 1.1 Positive Definite Functions . . . 10
  - 1.1.1 Examples of strictly positive definite RBF . . . 13
- 1.2 Conditionally Positive Definite Function . . . 15
  - 1.2.1 Examples of conditionally positive definite functions . . . 18
- 1.3 Reproducing Kernel Hilbert Space . . . 20
  - 1.3.1 Examples of Native Spaces . . . 22

2 Radial Basis Function Networks **25**
- 2.1 Historical remark . . . 26
- 2.2 Neural networks structure . . . 27
  - 2.2.1 Architecture . . . 27
  - 2.2.2 Learning Phase . . . 28
- 2.3 Radial Basis Function Networks . . . 29

3 Simultaneous approximation of function and its derivatives **35**
- 3.1 Direct and Indirect Methods . . . 35
- 3.2 Numerical Experiments with other RBF . . . 39
  - 3.2.1 Multiquadrics . . . 39
  - 3.2.2 Thin Plate Splines . . . 43
  - 3.2.3 Wendland . . . 43
  - 3.2.4 Considerations on Native spaces . . . 50
  - 3.2.5 Network existence and localization properties . . . 56

4 Solving high order differential equations using RBF networks **59**
- 4.1 Analysis of the condition number . . . 62
  - 4.1.1 Scaling the domain . . . 64
  - 4.1.2 Riley method and iterative methods . . . 65

5 Application to biological ODEs systems **67**
- 5.1 Reproducing Kernel Hilbert Spaces of vector-valued functions . . . 67
  - 5.1.1 Interpolation problem for vector-valued functions . . . 69
- 5.2 Solution of ODEs systems . . . 73
  - 5.2.1 Biological Model . . . 76
- 5.3 Conclusions . . . 79
  - 5.3.1 Future directions . . . 80

A DRBFN and IRBFN in two dimensions **81**

B A GUI interface with Matlab **85**

Bibliography **87**

Index **89**

Chapter 1

Radial Basis Functions: essential theory and properties

In this first chapter we summarize the main definitions and results about Radial Basis Functions (we can shorten in RBF) in the case of scalar-valued data. All the details on the Radial Basis Functions theory and theorems proofs are in [1].
Before giving the formal definition of a Radial Basis Function, we briefly introduce the problem of scattered data interpolation.

Problem 1.1 *Given data* $(\mathbf{x}_j, y_j),\ j = 1, ..., N$, *with* $\mathbf{x}_j \in \mathbb{R}^s$, $y_j \in \mathbb{R}$, *find a continuous function* P_f *such that* $P_f(\mathbf{x}_j) = y_j,\ j = 1, ..., N$.

One common approach to solve the scattered data interpolation problem is to assume that the function P_f is a linear combinations of certain basis function B_k.

$$P_f(\mathbf{x}) = \sum_{k=1}^{N} c_k B_k(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^s \tag{1.1}$$

This leads to a system of linear equations of the form

$$A\mathbf{c} = \mathbf{y} \tag{1.2}$$

where the entries of the interpolation matrix are $A_{jk} = B_k(\mathbf{x}_j),\ j, k = 1, ..., N$ and $\mathbf{c} = [c_1, ..., c_N]^T$, $\mathbf{y} = [y_1, ..., y_N]^T$.
In the univariate setting Problem 1 is always solvable. In the multivariate setting, on the contrary, the following result holds:

Theorem 1.2 *If* $\Omega \subset \mathbb{R}^s$, $s \geq 2$, *contains an interior point, then there exist no Haar spaces of continuous functions except for one-dimensional ones.*

This theorem tells us that if we want to have a well-posed multivariate scattered data interpolation problem we can no longer fix in advance the set of basis functions we plan to use for interpolation of arbitrary scattered data.
As an example in $I = [0, 1]$, we can construct the piecewise linear spline interpolant assuming P_f of the form

$$P_f(x) = \sum_{k=1}^{N} c_k |x - x_k|, \qquad x \in [0, 1] \tag{1.3}$$

and then determine the coefficients c_k imposing the interpolation conditions

$$P_f(x_j) = f(x_j) \quad j = 1, ..., N. \tag{1.4}$$

The points $\mathbf{x}_k$, to which the basic function is shifted, are usually referred as centers. While the points in which the function is computed or sampled are the data sites. There are no motivations in choosing the centers different from the data sites, so usually they coincide. Since the functions B_k are radially symmetric about their centers $\mathbf{x}_k$, this constitutes the first example of radial basis function.
We can generalize this approach to the interval $[0, 1]^s$ considering the Euclidean norm of the shift. In this case the interpolation matrix is:

$$\begin{pmatrix} \|\mathbf{x}_1 - \mathbf{x}_1\| & \|\mathbf{x}_1 - \mathbf{x}_2\| & \|\mathbf{x}_1 - \mathbf{x}_3\| & \cdots & \|\mathbf{x}_1 - \mathbf{x}_N\| \\ \|\mathbf{x}_2 - \mathbf{x}_1\| & \|\mathbf{x}_2 - \mathbf{x}_2\| & \|\mathbf{x}_2 - \mathbf{x}_3\| & \cdots & \|\mathbf{x}_2 - \mathbf{x}_N\| \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \|\mathbf{x}_N - \mathbf{x}_1\| & \|\mathbf{x}_N - \mathbf{x}_2\| & \|\mathbf{x}_N - \mathbf{x}_3\| & \cdots & \|\mathbf{x}_N - \mathbf{x}_N\| \end{pmatrix}$$

This matrix is called *distance matrix*.
In the following we give the formal definition of RBF.

Definition 1.1 *A function $\Phi : \mathbb{R}^s \to \mathbb{R}$ is called radial provided there exists a univariate function $\varphi : [0, \infty) \to \mathbb{R}$ such that*

$$\Phi(\mathbf{x}) = \varphi(r), \qquad r = \|\mathbf{x}\| \tag{1.5}$$

where $\|\cdot\|$ is some norm on $\mathbb{R}^s$, usually the Euclidean norm.

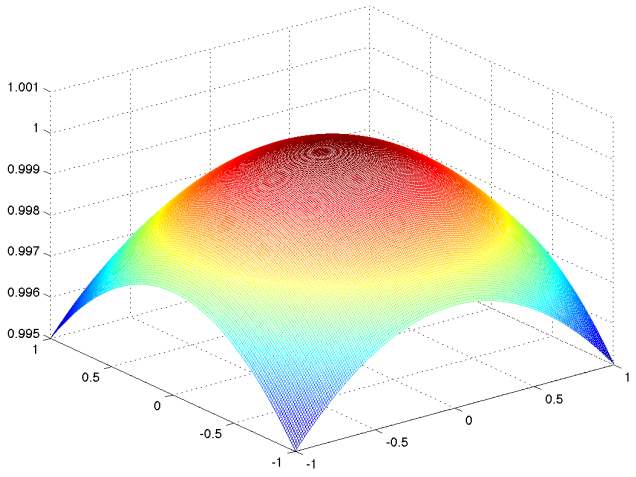
From the definition above we can observe that Φ is radially symmetric about its center. Indeed, if $\|\mathbf{x}_1\| = \|\mathbf{x}_2\|$ then $\Phi(\mathbf{x}_1) = \Phi(\mathbf{x}_2)$ where $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^s$. It follows that the simplest example of radial basis function is the Euclidean distance we used to solve the interpolation problem, i.e. $\varphi(r) = r$.
Now, after these considerations we can write the interpolation matrix as

$$\begin{pmatrix} \varphi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \varphi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \varphi(\|\mathbf{x}_1 - \mathbf{x}_3\|) & \cdots & \varphi(\|\mathbf{x}_1 - \mathbf{x}_N\|) \\ \varphi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \varphi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \varphi(\|\mathbf{x}_2 - \mathbf{x}_3\|) & \cdots & \varphi(\|\mathbf{x}_2 - \mathbf{x}_N\|) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varphi(\|\mathbf{x}_N - \mathbf{x}_1\|) & \varphi(\|\mathbf{x}_N - \mathbf{x}_2\|) & \varphi(\|\mathbf{x}_N - \mathbf{x}_3\|) & \cdots & \varphi(\|\mathbf{x}_N - \mathbf{x}_N\|) \end{pmatrix}$$

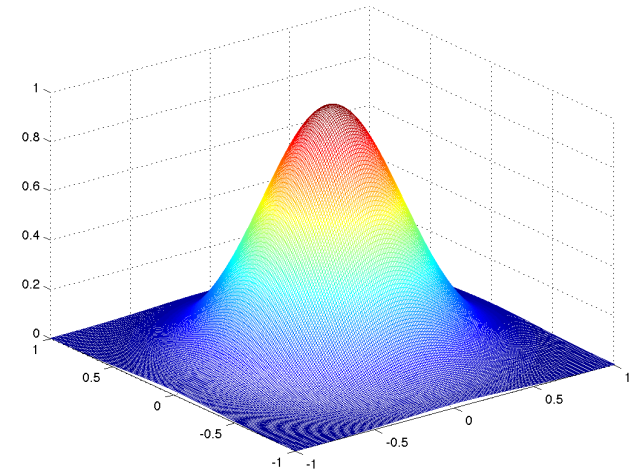
Example. The most important example of radial basis function is the *Gaussian*.

$$\varphi(r) = e^{-(\epsilon r)^2}, \quad r \in \mathbb{R} \tag{1.6}$$

where ϵ is a *shape parameter* and is related to the variance σ^2 of the normal distribution function by $\epsilon^2 = \frac{1}{2\sigma^2}$. In particular, a smaller value of ϵ causes the function to become flatter, while an increasing ϵ leads to a more peaked RBF (see figure 1.1 in the case $\epsilon = 2$).



(a) Gaussian with $\epsilon = 0.05$



(b) Gaussian with $\epsilon = 2$

Figure 1.1: Gaussian for different values of the shape parameter

If we compose the Gaussian with the euclidean norm we obtain, for any fixed center $\mathbf{x}_k$, a multivariate function:

$$\Phi_k(\mathbf{x}) = e^{-\epsilon^2 \|\mathbf{x}-\mathbf{x}_k\|}, \quad \mathbf{x} \in \mathbb{R}^s \tag{1.7}$$

Clearly, the connection between Φ and φ is given by

$$\Phi_k(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{x}_k\|). \tag{1.8}$$

Before talking about different radial basis functions, we introduce the *fill distance* which is a measure of the data distribution:

$$h = h_{X,\Omega} = \sup_{\mathbf{x}\in\Omega} \min_{\mathbf{x}_j \in X} \|\mathbf{x} - \mathbf{x}_j\|_2. \tag{1.9}$$

1.1 Positive Definite Functions

We have seen that the scattered data interpolation problem leads to the solution of a system of linear equations with matrix A.
It is well known, that the system has a unique solution whenever the matrix A is non-singular. Clearly this happens if the matrix is *positive definite*.

Definition 1.2 *A real symmetric matrix A is called positive semi-definite if its associated quadratic form is non-negative, i.e.,*

$$\sum_{j=1}^{N} \sum_{k=1}^{N} c_j c_k A_{jk} \geq 0 \tag{1.10}$$

for all $\mathbf{c} = [c_1, ..., c_N]^T \in \mathbb{R}^N$.
If the quadratic form is zero only for $\mathbf{c} \equiv 0$ then A is called positive definite.

All the eigenvalues of a positive definite matrix are positive, so the determinant, which is the product of eigenvalues, is always positive and the matrix is non-singular.
What is more, if we have basis functions B_k that generate a positive definite interpolation matrix, we will always have a well-posed interpolation problem.
To this end we introduce the concept of a positive definite function.

Definition 1.3 *A complex-valued continuous function $\Phi : \mathbb{R}^s \to \mathbb{C}$ is called positive definite on $\mathbb{R}^s$ if*

$$\sum_{j=1}^{N} \sum_{k=1}^{N} c_j \bar{c}_k \Phi(\mathbf{x}_j - \mathbf{x}_k) \geq 0 \tag{1.11}$$

for any N pairwise different points $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^s$ and $\mathbf{c} = [c_1, ..., c_N]^T \in \mathbb{C}^N$.
The function Φ is called strictly positive definite on $\mathbb{R}^s$ if the quadratic form (1.11) is zero only for $\mathbf{c} \equiv 0$.

We can now illustrate some of the most important properties and characterizations of positive definite functions. Proofs of these properties can be found in [1].

Theorem 1.3 *Some basic properties of positive definite functions are:*

1. *Non-negative finite linear combinations of positive definite functions are positive definite. If $\Phi_1, ..., \Phi_N$ are positive definite on $\mathbb{R}^s$ and $c_j \geq 0$, $j = 1, ..., N$, then*

$$\Phi(\mathbf{x}) = \sum_{j=1}^{N} c_j \Phi_j(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^s \tag{1.12}$$

   is also positive definite. Moreover, if at least one of the Φ_j is strictly positive definite and the corresponding $c_j > 0$, then Φ is strictly positive definite.

2. $\Phi(\mathbf{0}) \geq 0$.

3. $\Phi(-\mathbf{x}) = \overline{\Phi(\mathbf{x})}$.

4. *Any positive definite function is bounded. In fact,*

$$|\Phi(\mathbf{x})| \leq \Phi(\mathbf{0}). \tag{1.13}$$

5. *If Φ is positive definite with $\Phi(\mathbf{0}) = 0$ then $\Phi \equiv 0$.*

6. *The product of (strictly) positive definite functions is (strictly) positive definite.*

The definition and the properties above suggest that we should use strictly positive definite functions as basis functions in the interpolation problem.
In most of our work we will consider scalar-valued functions.
In the following we provide a characterization for real-valued (strictly) positive definite functions.

Definition 1.4 *A real-valued continous function Φ is positive definite on $\mathbb{R}^s$ if and only if it is even and*

$$\sum_{j=1}^{N} \sum_{k=1}^{N} c_j c_k \Phi(\mathbf{x}_j - \mathbf{x}_k) \geq 0 \tag{1.14}$$

for any N pairwise different points $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^s$ and $\mathbf{c} = [c_1, ..., c_N]^T \in \mathbb{C}^N$.
The function Φ is called strictly positive definite on $\mathbb{R}^s$ if the quadratic form above is zero only for $\mathbf{c} \equiv 0$.

We can now give the most famous characterization of positive definite functions in terms of Fourier transforms established by Bochner in 1932.

Theorem 1.4 *A (complex-valued) function $\Phi \in C(\mathbb{R}^s)$ is positive definite on $\mathbb{R}^s$ if and only if it is the Fourier transform of a finite non-negative Borel measure μ on $\mathbb{R}^s$, i.e.*

$$\Phi(\mathbf{x}) = \hat{\mu}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} e^{-i\mathbf{x}^T \cdot \mathbf{y}} d\mu(\mathbf{y}), \qquad \mathbf{x} \in \mathbb{R}^s \tag{1.15}$$

Proof We will prove only the easy direction, i.e. we will prove that assuming Φ the Fourier transform of a finite non-negative Borel measure then Φ is positive definite.
Thus,

$$\begin{aligned}\sum_{j=1}^{N}\sum_{k=1}^{N} c_j\bar{c}_k\Phi(\mathbf{x}_j-\mathbf{x}_k) &= \frac{1}{\sqrt{(2\pi)^s}}\sum_{j=1}^{N}\sum_{k=1}^{N}\left[c_j\bar{c}_k\int_{\mathbb{R}^s} e^{-i(\mathbf{x}_j-\mathbf{x}_k)^T\cdot\mathbf{y}}d\mu(\mathbf{y})\right]\\ &= \frac{1}{\sqrt{(2\pi)^s}}\int_{\mathbb{R}^s}\left[\sum_{j=1}^{N} c_j e^{-i\mathbf{x}_j^T\cdot\mathbf{y}}\sum_{k=1}^{N}\bar{c}_k e^{i\mathbf{x}_k^T\cdot\mathbf{y}}\right]d\mu(\mathbf{y})\\ &= \frac{1}{\sqrt{(2\pi)^s}}\int_{\mathbb{R}^s}\Big|\sum_{j=1}^{N} c_j e^{-i\mathbf{x}_j^T\cdot\mathbf{y}}\Big|^2 d\mu(\mathbf{y}) \geq 0\end{aligned}$$

where the last inequality holds because of the assumptions imposed on the measure μ.

□

To guarantee that the interpolation problem will be well-posed we extend Bochner 's theorem to the case of *strictly* positive definite functions.
To obtain this result we should introduce the concept of *carrier* of a (non-negative) Borel measure defined on some topological space X:

$$X \setminus \cup\{O : O \text{ is open and } \mu(O) = 0\} \tag{1.16}$$

Theorem 1.5 *Let μ be a non-negative finite Borel measure on $\mathbb{R}^s$ whose carrier is a set of nonzero Lebesgue measure. Then the Fourier transform of μ is strictly positive definite on $\mathbb{R}^s$.*

Proof Theorem's proof proceeds exactly as in the Bochner's theorem.
Thus, we obtain that the Fourier transform is positive definite. To show that is strictly positive definite, let

$$g(\mathbf{y}) = \sum_{j=1}^{N} c_j e^{-i\mathbf{x}\cdot\mathbf{y}} \tag{1.17}$$

and assume that the points $\mathbf{x}_j$ are all distinct and $\mathbf{c} \neq 0$.
In this case the functions $\mathbf{y} \to e^{-i\mathbf{x}\cdot\mathbf{y}}$ are linearly independent, so that $g \neq 0$. Since g is an entire function, its zero set can have no accumulation point and therefore it has Lebesgue measure zero.
Now, the only remaining way to make the above inequality an equality is if the carrier of μ is contained in the zero set of g, i.e. has Lebesgue measure zero.
But this is ruled out in the hypothesis of the theorem. □

Finally, we state a criterion to check whether a given function is strictly positive definite:

Theorem 1.6 *Let Φ be a continuous function in $L_1(\mathbb{R}^s)$. Φ is strictly positive definite if and only if Φ is bounded and its Fourier transform is non-negative and not identically equal to zero.*

Till now we haven't required the function Φ to be radial. In fact, the function

$$P_f(\mathbf{x}) = \sum_{k=1}^{N} c_k \Phi(\mathbf{x} - \mathbf{x}_k), \quad \mathbf{x} \in \mathbb{R}^s \tag{1.18}$$

will yield an interpolant that is only translation invariant. If we want invariance also under rotations and reflections we should use positive definite functions that are radial on $\mathbb{R}^s$.

Lemma 1.1 *If $\Phi = \varphi(\|\cdot\|)$ is (strictly) positive definite and radial on $\mathbb{R}^s$ then Φ is also (strictly) positive definite and radial on $\mathbb{R}^\sigma$ for any $\sigma \leq s$.*

1.1.1 Examples of strictly positive definite RBF

In this section we give some examples of strictly positive definite radial functions which we will use later for our purposes. To show that they are strictly positive definite we will use the criterion in Theorem (1.6).

- **Gaussians**:

$$\Phi(\mathbf{x}) = e^{-\epsilon^2 \|\mathbf{x}\|^2} \quad \epsilon > 0 \tag{1.19}$$

  The Fourier transform of this function is a Gaussian and this explains why the function is strictly positive definite.

$$\hat{\Phi}(\boldsymbol{\omega}) = \frac{1}{(\sqrt{2}\epsilon)^s} e^{-\frac{\|\boldsymbol{\omega}\|^2}{4\epsilon^2}} \tag{1.20}$$

- **Matérn Functions**

$$\Phi(\mathbf{x}) = \frac{K_{\beta - \frac{s}{2}}(\|\mathbf{x}\|)\|\mathbf{x}\|^{\beta - \frac{s}{2}}}{2^{\beta-1}\Gamma(\beta)} \quad \beta > \frac{s}{2} \tag{1.21}$$

  where K_ν is the modified Bessel function of the second kind of order ν.

  The Fourier transform of the Matérn functions is given by the Bessel kernels

$$\hat{\Phi}(\boldsymbol{\omega}) = (1 + \|\boldsymbol{\omega}\|^2)^{-\beta} > 0 \tag{1.22}$$

  Therefore the Matérn functions are strictly positive definite on $\mathbb{R}^s$ $\forall s < 2\beta$.

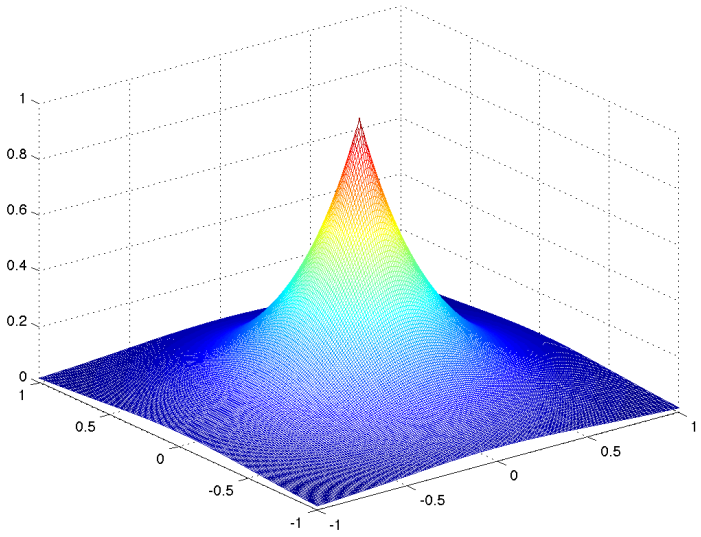


Figure 1.2: Matérn with $\beta = \frac{s+1}{2}$

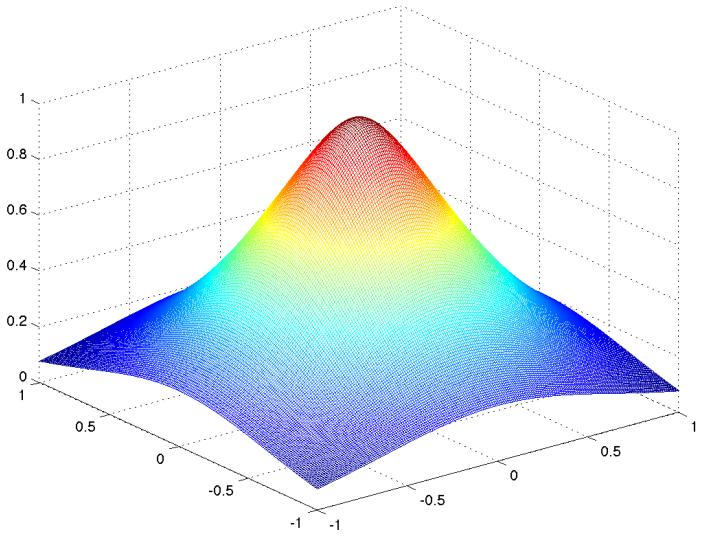


Figure 1.3: Matérn with $\beta = \frac{s+3}{2}$

- **Generalized Inverse Multiquadrics**

$$\Phi(x) = (1 + \|\mathbf{x}\|^2)^{-\beta}, \quad \beta > \frac{s}{2} \tag{1.23}$$

This is exactly the Fourier transform of Matérn function.

To show that is positive definite we need Hankel inversion theorem, which states that the Fourier transform for radial functions is its own inverse, i.e.

$$\mathcal{F}_s[\mathcal{F}_s\varphi] = \varphi \tag{1.24}$$

Thus, using Hankel's theorem, we can reverse the role of the function and of the Fourier transform in the previous example and we obtain that generalized inverse multiquadrics are strictly positive definite for $s < 2\beta$.

When $\beta = \frac{1}{2}$ we have the particular case of Hardy's inverse multiquadric.

1.2 Conditionally Positive Definite Function

Definition 1.5 *A real symmetric matrix A is called conditionally positive semidefinite of order one if its associated quadratic form is non-negative, i.e.*

$$\sum_{j=1}^{N}\sum_{k=1}^{N} c_j c_k A_{jk} \geq 0 \tag{1.25}$$

for all $\mathbf{c} = [c_1, ..., c_N] \in \mathbb{R}^N$ *that satisfy*

$$\sum_{j=1}^{N} c_j = 0 \tag{1.26}$$

If $\mathbf{c} \neq 0$ *implies strict inequality in (1.25) then A is called conditionally positive definite of order one.*

The reason to introduce conditionally positive definite matrices (and later conditionally positive definite functions) rises from the fact that sometimes is desirable to have an interpolant which exactly reproduces simple polynomial functions.
For example if the data are constants or come from a linear function, it would be nice if our interpolant were also constant or linear, respectively.
The problem is that the methods we have seen so far do not reproduce even simple polynomial functions.
A simple remedy to this problem is to modify the assumptions on the form of the interpolant adding a polynomial basis.
This leads to the following formulation of the interpolation problem:

$$P_f(\mathbf{x}) = \sum_{k=1}^{N} c_k \varphi(\|\mathbf{x} - \mathbf{x}_k\|) + \sum_{l=1}^{M} d_l p_l(\mathbf{x}) \qquad \mathbf{x} \in \mathbb{R}^s \tag{1.27}$$

where $p_1, ..., p_M$ form a basis for the $M = \binom{m-1+s}{m-1}$- dimensional linear space Π^s_{m-1} of polynomials of total degree less than or equal to $m-1$ in s variables.
To obtain a square matrix and thus a unique solution we also add the following conditions

$$\sum_{k=1}^{N} c_k p_l(\mathbf{x}_k) = 0 \tag{1.28}$$

Thus we obtain the following augmented system:

$$\begin{pmatrix} A & P \\ P^T & O \end{pmatrix} \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} y \\ 0 \end{pmatrix}$$

where $A_{jk} = \varphi(\|\mathbf{x}_j - \mathbf{x}_k\|),\ j,k = 1, ..., N,\ P_{jl} = p_l(\mathbf{x}_k)\ k = 1, ..., N,\ l = 1, ..., M$, $\mathbf{c} = [c_1, ..., c_N]^T$, $\mathbf{d} = [d_1, ..., d_M]^T$, $\mathbf{y} = [y_1, ..., y_N]^T$, $\mathbf{0}$ is a zero vector of length M and O is a $M \times M$ zero matrix.
Now we need conditions for the augmented matrix to be non-singular:

Theorem 1.7 *Let A be a real symmetric $N \times N$ matrix that is conditionally positive definite of order one, and let $P = [1, ..., 1]^T$ be a $N \times 1$ matrix.*
Then the system of linear equations

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ d \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} \tag{1.29}$$

is uniquely solvable.

We can observe from the previous theorem that since strictly positive definite matrices are also conditionally positive definite of order one, the (augmented) radial basis function interpolation matrix for reproduction of polynomials of order 0 (i.e. scalar constants) is non-singular. This means that strictly positive definite matrices reproduce exactly constants.
Now, we can generalize the results above introducing conditionally positive definite and strictly conditionally positive definite *functions* of order m.
Clearly, these functions provide the natural generalization of RBF interpolation with polynomial reproduction.

Definition 1.6 *A complex-valued continous function Φ is called conditionally positive definite of order m on $\mathbb{R}^s$ if*

$$\sum_{j=1}^{N} \sum_{k=1}^{N} c_j \bar{c}_k \Phi(\mathbf{x}_j - \mathbf{x}_k) \geq 0 \tag{1.30}$$

for any N pairwise distinct points $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^s$ and $\mathbf{c} = [c_1, ..., c_N]^T \in \mathbb{C}^N$ satisfying

$$\sum_{j=1}^{N} c_j p(\mathbf{x}_j) = 0 \tag{1.31}$$

for any complex-valued polynomial p of degree at most $m-1$.
The function Φ is called strictly conditionally positive definite of order m on $\mathbb{R}^s$ if the quadratic form above is zero only for $\mathbf{c} \equiv 0$.

The following lemma gives an immediate property of conditionally positive definite functions.

Lemma 1.2 *A function that is (strictly) conditionally positive definite of order m on $\mathbb{R}^s$ is also (strictly) conditionally positive definite of any higher order. In particular, a (strictly) positive definite function is always (strictly) conditionally positive definite of any order.*

As we have done for (strictly) positive definite functions, we can give a characterization also for scalar-valued functions.

Definition 1.7 *A real-valued continous even function Φ is called conditionally positive definite of order m on $\mathbb{R}^s$ if*

$$\sum_{j=1}^{N} \sum_{k=1}^{N} c_j c_k \Phi(\mathbf{x}_j - \mathbf{x}_k) \geq 0 \tag{1.32}$$

for any N pairwise distinct points $\mathbf{x}_1, ..., \mathbf{x}_N \in \mathbb{R}^s$ and $\mathbf{c} = [c_1, ..., c_N]^T \in \mathbb{C}^N$ satisfying

$$\sum_{j=1}^{N} c_j p(\mathbf{x}_j) = 0 \tag{1.33}$$

for any real-valued polynomial p of degree at most $m-1$.
The function Φ is called strictly conditionally positive definite of order m on $\mathbb{R}^s$ if the quadratic form above is zero only for $\mathbf{c} \equiv 0$.

Now we can state the generalization of theorem 1.7 for strictly conditionally positive definite functions of order m.

Theorem 1.8 *If the real-valued even function Φ is strictly conditionally positive definite of order m on $\mathbb{R}^s$ and the points $\mathbf{x}_1, ..., \mathbf{x}_N$ form an $(m-1)$-unisolvent set, then the system of linear equations (1.29) is uniquely solvable.*

Where a set is said to be $(m-1)$-unisolvent if the only polynomial of total degree at most $m-1$ interpolating zero data is the zero polynomial.
We can also give an integral representation for conditionally positive definite functions in terms of the Fourier transform.

Theorem 1.9 *Suppose the complex-valued function Φ owns a generalized Fourier transform $\hat{\Phi}$ of order m which is continous on $\mathbb{R}^s \setminus \{0\}$. Then Φ is strictly conditionally positive definite of order m if and only if $\hat{\Phi}$ is non-negative and non-vanishing.*

1.2.1 Examples of conditionally positive definite functions

We list some examples of strictly conditionally positive definite functions that will be useful later for our purpose.

- **Generalized Multiquadrics**

  The Multiquadric Radial Basic Function

  $$\Phi(\mathbf{x}) = (1 + \|\mathbf{x}\|^2)^{\beta}, \quad \mathbf{x} \in \mathbb{R}^s, \beta \in \mathbb{R} \setminus \mathbb{N}_0 \tag{1.34}$$

  was used for the first time, to solve an interpolation problem, in 1968 by Iowa State University Geodesist Roland Hardy. He then described his method in a paper published in 1971 ([2]).

  His study was motivated by a cartography problem who had tried to solve using Trigonometric and Algebraic interpolation methods, both of which were unsatisfactory.

  The method remained unnoticed till 1979 when Richard Franke, in a study done at the Naval Postgraduate School, concluded that Hardy 's interpolation method was the best.

  What is more, he conjectured that the system matrix of the method was invertible and that the method was well posed.

  The matrix invertibility was proved in 1986 by Charles Micchelli.

  Now, we can prove that generalized multiquadrics are strictly conditionally positive definite using generalized Fourier transform:

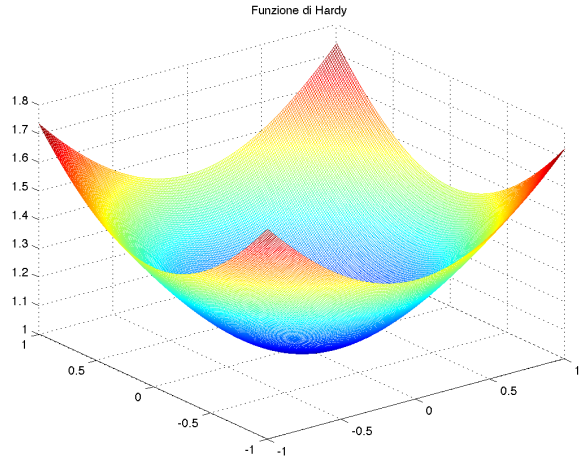
  $$\hat{\Phi}(\boldsymbol{\omega}) = \frac{2^{1+\beta}}{\Gamma(-\beta)} \|\boldsymbol{\omega}\|^{-\beta-\frac{s}{2}} K_{\beta+\frac{s}{2}}(\|\boldsymbol{\omega}\|), \quad \boldsymbol{\omega} \neq 0 \tag{1.35}$$

  This transform is of order $m = \max(0, \lceil \beta \rceil)$.

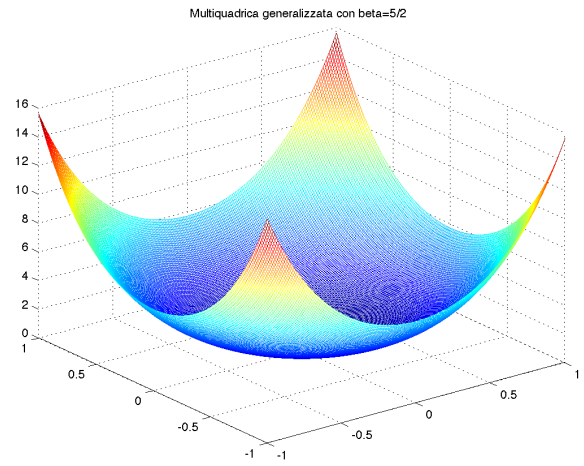
  As before, K_ν is the modified Bessel function of the second kind of order ν.

  Since the generalized Fourier transform is positive with a singularity of order m in the origin, generalized multiquadrics are strictly conditionally positive definite of order $m = \lceil \beta \rceil$ (and higher).

  If $\beta < 0$ the Fourier transform is the classical one and we obtain again the inverse generalized multiquadrics of the previous section that are strictly conditionally positive definite of order $m = 0$ that means they are strictly positive definite.



(a) Hardy's Multiquadrics obtained with $\beta = \frac{1}{2}$



(b) Multiquadric with $\beta = \frac{5}{2}$

- **Thin Plate Splines**

$$\Phi(\mathbf{x}) = \|\mathbf{x}\|^{2\beta} \log(\|\mathbf{x}\|), \quad \mathbf{x} \in \mathbb{R}^s, \beta \in \mathbb{N} \tag{1.36}$$

This function has generalized Fourier transform

$$\hat{\Phi}(\boldsymbol{\omega}) = (-1)^{\beta+1} 2^{2\beta-1+\frac{s}{2}} \Gamma(\beta + \frac{s}{2}) \beta! \|\boldsymbol{\omega}\|^{-s-2\beta} \tag{1.37}$$

of order $m = \beta + 1$. Therefore the Thin Plate Splines are strictly conditionally positive definite of order $m = \beta + 1$.

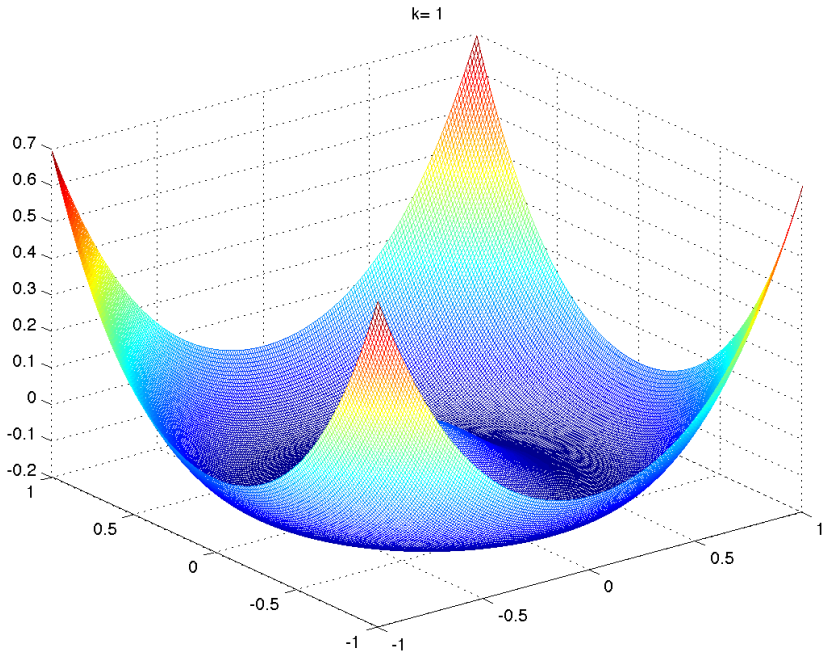


Figure 1.4: Thin Plate Spline with $\beta = 1$

1.3 Reproducing Kernel Hilbert Space

In this section we associate with each (strictly positive definite) radial basic function a certain space of functions called its *native space*. This gives a connection with the concept of Reproducing Kernel Hilbert Space (RKHS).
We will generalize this theory in the last chapter for the vectorial case where we will study the problem of interpolate a vector-valued function.
The following statement is a formal definition:

Definition 1.8 *Let $\mathcal{H}$ be a real Hilbert space of function $f : \Omega(\subset \mathbb{R}^s) \to \mathbb{R}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. A function $K : \Omega \times \Omega \to \mathbb{R}$ is called reproducing kernel for $\mathcal{H}$ if*

1. $K(\cdot, \mathbf{x}) \in \mathcal{H} \quad \forall \mathbf{x} \in \Omega$
2. $f(\mathbf{x}) = \langle f, K(\cdot, \mathbf{x}) \rangle_{\mathcal{H}} \quad \forall f \in \mathcal{H} \quad$ *and* $\quad \forall \mathbf{x} \in \Omega$

The name reproducing kernel is inspired by the reproducing property 2 in the definition (1.8).
A RKHS is unique and satisfies the following further properties:

Theorem 1.10 *Suppose $\mathcal{H}$ is a Hilbert space of functions $f : \Omega \to \mathbb{R}$ with reproducing kernel K. Then we have:*

1. $K(\mathbf{x},\mathbf{y}) = \langle K(\cdot,\mathbf{y}), K(\cdot,\mathbf{x})\rangle_{\mathcal{H}} \qquad \forall \mathbf{x},\mathbf{y} \in \Omega$
2. $K(\mathbf{x},\mathbf{y}) = K(\mathbf{y},\mathbf{x}) \qquad for\, \mathbf{x},\mathbf{y} \in \Omega$
3. *if* $\|f - f_n\|_H \to 0 \quad for\ n \to \infty \quad then\ |f(\mathbf{x}) - f_n(\mathbf{x})| \to 0 \quad \forall \mathbf{x} \in \Omega$
   The last property means that convergence in Hilbert space norm implies pointwise convergence.

From Riesz representation theorem we have that existence of a reproducing kernel is equivalent to the fact that the point evaluation functionals δ_x are bounded in Ω. This means that there exists a positive constant M depending on x such that

$$|\delta_{\mathbf{x}} f| = |f(\mathbf{x})| \leq M\|f\|_{\mathcal{H}} \quad \forall f \in \mathcal{H}, \quad \mathbf{x} \in \Omega \tag{1.38}$$

This observation is very useful because gives us a connection with the vectorial case where the same theory is valid. Indeed, we will see in the next chapters, that reproducing kernel in the multidimensional case satisfies almost the same properties of the scalar case.
Now we would find a connection between the reproducing kernel, which is known to be positive definite, and strictly positive definite function.
One direction of this connection is given by the following theorem:

Theorem 1.11 *Suppose $\mathcal{H}$ is a RKHS with reproducing kernel $K : \Omega \times \Omega \to \mathbb{R}$. Then K is positive definite. Moreover K is strictly positive definite if and only if the point evaluation functionals $\delta_{\mathbf{x}}$ are linearly independent in $\mathcal{H}^*$.*

Now we show that every strictly positive definite radial basic function can be associated with a reproducing kernel Hilbert space, its native space.
Indeed, for all the functions f in the space $\mathcal{H}$, we can write:

$$f = \sum_{j=1}^{N} c_j K(\cdot,\mathbf{x}_j) \quad \mathbf{x}_j \in \Omega \tag{1.39}$$

As a consequence of Theorem 1.10 we have that

$$\begin{aligned}
\|f\|^2_{\mathcal{H}} &= \langle f, f\rangle_{\mathcal{H}} \\
&= \langle \sum_{j=1}^{N} c_j K(\cdot,\mathbf{x}_j), \sum_{k=1}^{N} c_k K(\cdot,\mathbf{x}_k)\rangle_{\mathcal{H}} \\
&= \sum_{j=1}^{N}\sum_{k=1}^{N} c_j c_k \langle K(\cdot,\mathbf{x}_j), K(\cdot,\mathbf{x}_k)\rangle_{\mathcal{H}} \\
&= \sum_{j=1}^{N}\sum_{k=1}^{N} c_j c_k K(\mathbf{x}_j,\mathbf{x}_k).
\end{aligned}$$

So, we can define the (possibly infinite-dimensional) space

$$\mathcal{H}(\Omega) = span\,\{K(\cdot, \mathbf{y}) : \mathbf{y} \in \Omega\} \tag{1.40}$$

with the associated bilinear form $\langle \cdot, \cdot \rangle_K$ given by

$$\langle \sum_{j=1}^{N_K} c_j K(\cdot, \mathbf{x}_j), \sum_{k=1}^{N_K} d_k K(\cdot, \mathbf{y}_k) \rangle_K = \sum_{j=1}^{N_K} \sum_{k=1}^{N_K} c_j d_k K(\mathbf{x}_j, \mathbf{y}_k). \tag{1.41}$$

where $N_K = \infty$ is also allowed.

Theorem 1.12 *If $K : \Omega \times \Omega \to \mathbb{R}$ is a symmetric strictly positive definite kernel, then the bilinear form $\langle \cdot, \cdot \rangle_K$ defines an inner product on $H_K(\Omega)$. Furthermore, $H_K(\Omega)$ is a pre-Hilbert space with reproducing kernel K.*

Since this space is not complete, we define the native space $N_K(\Omega)$ to be the completion of $H_K(\Omega)$ with respect to the K-norm $\| \cdot \|_K$ so that $\|f\|_K = \|f\|_{N_K(\Omega)}$ for all $f \in H_K(\Omega)$. From the construction above, we have that we can think $K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x} - \mathbf{y})$, where Φ is a strictly positive definite radial function.

1.3.1 Examples of Native Spaces

In this subsection we give some examples of native spaces associated to a class of radial basis function.

- Any strictly positive definite function Φ whose Fourier transform decays only algebraically has a Sobolev Space as its native space.

  In particular **Matérn** functions has $\mathcal{N}_{\Phi_\beta}(\mathbb{R}^s) = W_2^\beta(\mathbb{R}^s)$ with $\beta > \frac{s}{2}$ as native space.

  Where $W_2^m(\mathbb{R}^s) = \left\{ f \in L_2(\mathbb{R}^s) \cap C(\mathbb{R}^s) : \hat{f}(\cdot)(1 + \| \cdot \|_2^2)^{m/2} \in L_2(\mathbb{R}^s) \right\}$ is the Sobolev space.

  An equivalent definition for the set above is:

  $$W_2^m(\Omega) = \{ f \in L_2(\Omega) \cap C(\Omega) : D^\alpha f \in L_2(\Omega)\ \forall |\alpha| \leq m, \alpha \in \mathbb{N}^s \} \tag{1.42}$$

  Wendland's compactly supported functions, that we will introduce later, have native spaces $\mathcal{N}_{\Phi_{s,k}}(\mathbb{R}^s) = W_2^{s/2+k+1/2}(\mathbb{R}^s)$.

- the native spaces for **Gaussians** and **Inverse Multiquadrics** are rather small. Nevertheless, Gaussian 's native space contains the class of so-called band-limited functions, i.e. functions whose Fourier transform is compactly supported.

  These functions play an important role in sampling theory where the main result is Shannon's theorem:

Theorem 1.13 *Suppose $f \in C(\mathbb{R}^s) \cap L_1(\mathbb{R}^s)$ such that its Fourier transform vanishes outside the cube $Q = [-\frac{1}{2}, \frac{1}{2}]^s$. Then f can be uniquely reconstructed from its values on $\mathbb{Z}^s$, i.e.*

$$f(x) = \sum_{\boldsymbol{\xi} \in \mathbb{Z}^s} f(\xi) sinc(\mathbf{x} - \boldsymbol{\xi}), \quad \mathbf{x} \in \mathbb{R}^s \tag{1.43}$$

In the statement above $sinc(\mathbf{x}) = \prod_{d=1}^{s} \frac{\sin(\pi \mathbf{x}_d)}{\pi \mathbf{x}_d}$.

- The construction of native spaces for conditionally positive definite functions is rather technical, so we limited the discussion to strictly conditionally positive definite functions.

  For example, for **Thin Plate Splines** the native space is the so-called *Beppo-Levi space* of order k:

$$BL_k = \{f \in C(\mathbb{R}^s) : D^\alpha f \in L_2(\mathbb{R}^s) \text{ for all } |\alpha| = k, \alpha \in \mathbb{N}^s\} \tag{1.44}$$

Chapter 2

Radial Basis Function Networks

An artificial neural network is a system, inspired to biological neural networks, which processes information and data.
Even if nowadays computers are more powerful and faster than some years ago, they are still not able to solve problems that are trivial for a human being, such as the recognition of a specific object.
This is due to the fact that the approach with which computers solve problems is by following a specific algorithm, i.e. a series of sequential instructions and operations, like the ones necessary to do complex computations or to store a big amount of data.
Artificial neural networks are composed by a lot of units, called neurons, some of which receive information from the environment (*input units*), some of which give responses (*output units*) and finally others which are hidden inside the networks (*hidden units*).

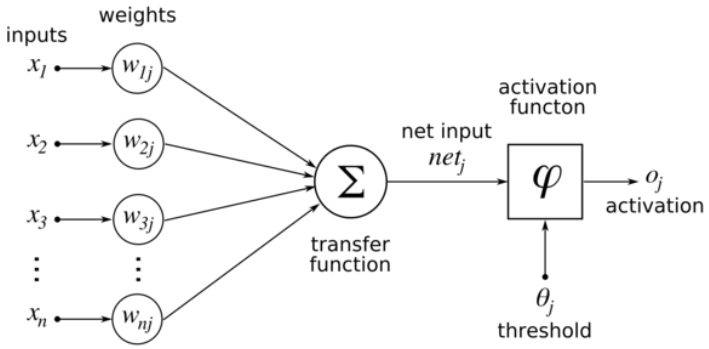


Figure 2.1: Artificial Neural Network

Every unit becomes active if the amount of information it has received is more than a fixed threshold and then it emits a signal which is transmitted along the network till it reaches other units.
Every connecting point between units can increase or decrease the signal as it happens with brain's synapse. For this reason these points are called synapse or even weight, since they weigh the intensity of the transmitted signal.
Furthermore, an artificial neural network *learns* which are the correct association input-output from the presentation of different examples. Usually the learning phase is quite long and it requires the examples to be shown a lot of times. During this phase the network changes his synaptic connections following some learning rules and adapting

itself to the problem is going to solve.

Thanks to this structure, a neural network, simulating biological neural networks, performs tasks collectively and in parallel. Its main characteristics are:

- *Robustness:* it is resistant against noise, since it gives a correct output even when the input is noisy or a part of the connections are destroyed. Clearly the higher the noise, the lower the accuracy. But the way the error increases is peculiar, it could increase uniformly or it could be present in response of some inputs and totally vanish for others. What is more, the advantage in using neural networks is that they can learn again and improve their performances.

- *Flexibility*: thanks to the *learning* phase, a neural network has a lot of application fields, for example: information theory (data compression, sonar signal recognition, alphanumeric character recognition), control systems, medical diagnosis (imaging analysis) and neuroscience.

  Moreover, since it learns from presentation of patterns, there is no need to know exactly the solution of a problem in all its details. It is sufficient to know the project's aim and some constraints to choose the neural network model in the right way.

- *Generalization*: In the learning phase a neural network tries to memorize all the common characteristics of every shown pattern in order to use this information when dealing with unknown patterns.

- *Recovery*: a neural network is able to recover data and information starting from some incomplete or corrupted data, in the same way human beings are able to remember something even if they have only partial information.

2.1 Historical remark

The first model of artificial neuron was created by Mc-Culloch and Pitts in the 1943. They proposed to assemble a lot of neurons to compute complex logic functions, but this first kind of artificial neural network was not able to learn and wasn't adaptive.
Some years later, the Canadian psychologist Donald Hebb introduced the concept of *learning* for artificial networks. His idea was that if two neurons are acting simultaneously then their connection value is augmented. Even if Debb's studies were only qualitative they set the path for further investigations.
In fact, during the fifties and sixties there was a big improvement mostly due to Frank Rosenblatt' s work. He created the first perceptron, a connection scheme of a neural network, and he developed also an iterative algorithm to correct the errors arising during

the solving process.
In the same period, Widrow and his student Hoff, developed a similar algorithm but more effective, called *delta rule* or ADALINE. The idea was to give a measure of the error and then to adapt the neurons connections consequently, estimating the gap between the neural network response and the exact response.

Afterwards, while in the second half of sixties Minsky and Papert analysis on perceptron model limitations reduced the initial enthusiasm towards artificial network, a renovated interest in this research field occurred in the eighties, with the worth work led by John Hopfield, who compared the functioning of a neural network to a physical system, and, in Japan, with the Fukushima work, who proposed a network for character recognition, later developed and modified.
The full affirmation of neural network field arrived in 1986, when the back-propagation algorithm was published. This study actually represented the reply to Minsky and Papert's criticism, since it gave a recursive method, based on the delta rule, to modify neural network synaptic values.
Since them till nowadays neural networks have known a great improvement and development in a lot of research fields.

2.2 Neural networks structure

In this section we analyse some common characteristics of a generic artificial neural network (For an extended treatment look at [3]). In the following sections we concentrate on Radial Basis Function Neural Networks (RBFN) [4].

2.2.1 Architecture

As we said in the introduction to this chapter, artificial networks are composed by units or neurons. Every neuron is characterized by an activation threshold and by an activation function.
The connection between two neurons is called synapse.
In the following we indicate with $\mathbf{x} = \{x_1, ..., x_j, ...x_N\}$ the input vector and with $\mathbf{y} = \{y_1, ..., y_i, ...y_n\}$ the output vector (we suppose they do not necessarily have the same cardinality), so that we can write the response of the i-th neuron as:

$$y_i = \Phi(A) = \Phi(\sum_{j=1}^{N} \omega_{ij} x_j - \theta_i) \tag{2.1}$$

where we have supposed $\mathbf{x}$ composed by N components and where Φ is the activation function, $\boldsymbol{\theta}$ is the activation threshold vector and $\boldsymbol{\omega}$ is the vector of weights associated

to every synapse. The weights' vector changes during the learning phase to adapt the network and can assume positive or negative values.
The activation function's choice determines the different responses of the net. We list some examples:

- **Staircase function**:

$$\Phi(A) = \begin{cases} 1 & A > \theta \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

$$\Phi(A) = \begin{cases} 1 & A > \theta \\ -1 & \text{otherwise} \end{cases} \tag{2.3}$$

  In both cases the neuron can transmit only a bit of information.

- **Linear function**

$$\Phi(A) = kA \tag{2.4}$$

- **Sigmoid function**

$$\Phi(A) = \frac{1}{1 + e^{-kA}} \tag{2.5}$$

  where the constant k controls the curve steepness; for instance if $k \to \infty$ then the sigmoid function tends to the staircase function.

For what concern network's architecture, what distinguishes a network from another is the number of input and output units. Using this as criterion, we can classify networks in *hetero-associative* or *auto-associative*.
In hetero-associative nets input units are completely distinct from the output vector in a way that, for example, they can differ in the number of units which compose input and output.
In auto-associative nets there is a unique layer of input-output such that all the units are connected together. This auto-referring feature entails the possibility of an instantaneous control of the exchanged information between input and output, and that's the reason why these nets are used to predict time dependent processes.

Another distinguishing feature is the number of hidden layers. The most common architecture is a *multi-layer* architecture in which there is more than one hidden layer. In this structure, usually, the information travel from the lowest layer to the output. This kind of transmission is called *feed-forward*.

2.2.2 Learning Phase

One important step in the use of an artificial neural network is the learning phase. We can make a distinction between *supervised learning*, where you have a measure of error based

on the difference between expected response and the actual output, and *unsupervised learning*.
In the first scheme, you exactly know the input pattern and the output pattern, thus this kind of learning involves different algorithms to measure the goodness of the response. In the latter, the main difference is that the desired response is not known or determined but there are a series of conditions the output should satisfy. This kind of network can learn from the environment and can use some a priori knowledge to find a solution to an unknown pattern with good approximation.
Anyway, all the learning algorithms have some common features:

1. the initial values of the weights are chosen randomly in a small range (for example among -0.1 and 0.1) or are set to zero.

2. The learning phase consists in the repeated presentation of training patterns. The weights are modified after the presentation of every pattern or after all the patterns have been shown. The new value of the weights is given by:

$$\omega_{ij}^t = \omega_{ij}^{t-1} + \Delta\omega_{ij}^t \tag{2.6}$$

   that means we have to determine only $\Delta\omega_{ij}^t$.

3. When the training is over, the weights' values are stored and one can test the network on unknown pattern (*Generalization phase*).

2.3 Radial Basis Function Networks

A Radial Basis Function Network is composed, as in the formulation given by Broomhead and Lowe [5], by an input layer, an output layer and by a unique hidden layer.
The mathematical transformation from the input layer to the hidden layer, i.e. the activation function Φ (see 3.5), is always non-linear, while the transformation from the hidden layer to the output (the function A) is linear. Thus, we are allowed to use radial basis functions as activation functions.
The motivation for choosing a non-linear activation function is that **the network can learn non-linearly separable transformations**. We give a short explanation of this statement in the following.

An example which involves non-linearly separable transformations is the problem of patterns classification. This means that, given a rule to classify (i.e. to distinguish a pattern from another) patterns, the network should be able to divide the input patterns into classes of similar patterns, on the base of the rule established from the beginning of the process.

The learning phase for this kind of problems consists in the creation of a separation vector for the input space, such that in the output every pattern is in its own class. If we imagine patterns as points in the space which are divided in two classes on the base of their coordinates, the problem of classification is to find a line or a plane or an hyperplane such that points in the first class lay on one side of the line or plane or hyperplane and points in the second class lay on the other side. If the separation line or plane or hyperplane exists, the problem is said to be *linearly separable.*
The main theorem on the separability problem is *Cover's Theorem*, which also provide a justification for taking a high-dimensional hidden unit.

Theorem 2.1 *"A complex pattern-classification problem, cast in a high-dimensional space non-linearly, is more likely to be linearly separable than in a low-dimensional space." (Cover 1965)*

We can state this theorem in a different way by saying :
Consider a pattern of points $\mathbf{x} \in X$, each of which is assigned to a region of the space X^+ or X^-. We called this subdivision a *dichotomy* (binary partition). We say that the dichotomy is separable if there exists a surface separating the two classes of points X^+ and X^-.

For every pattern $\mathbf{x} \in X$, consider now a set of real-valued functions

$$\{\varphi_i(\mathbf{x}) | i = 1, ..., m_2\} \qquad \text{such that } \varphi(\mathbf{x}) = [\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), ..., \varphi_{m_2}(\mathbf{x})]. \tag{2.7}$$

If the pattern vector lives in a m_1-dimensional space, every

$$\varphi(\mathbf{x}) = (\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), ..., \varphi_{m_2}(\mathbf{x})) \tag{2.8}$$

is a map from the m_1-dimensional input space into an m_2-dimensional space. Function φ_i is said to be an hidden function because it plays a role similar to the one a hidden unit plays in a feedforward neural network.

Definition 2.1 *A dichotomy $\{X^+, X^-\}$ of X is φ-separable if there exists an m_1-dimensional vector $\boldsymbol{w}$ such that we may write:*

$$\begin{cases} \boldsymbol{w}^T \varphi(\mathbf{x}) \geq 0 & \mathbf{x} \in X^+ \\ \boldsymbol{w}^T \varphi(\mathbf{x}) < 0 & \mathbf{x} \in X^- \end{cases} \tag{2.9}$$

The expression

$$\boldsymbol{w}^T \varphi(\mathbf{x}) = 0 \tag{2.10}$$

define the separating hyperplane in the hidden space. The inverse image of this hyperplane

$$\mathbf{x} : \boldsymbol{w}^T \varphi(\mathbf{x}) = 0 \tag{2.11}$$

defines the separating surface in the input space.

This means that if φ maps the input pattern into a high dimensional space, then the problem can become linearly separable.

Nevertheless, we have to observe that it is not always necessary to map the data in high dimensional space. Sometimes the use of a non-linear hidden function is sufficient to obtain the separability, as it is shown in the following example.
The most famous case of non-linear separation problem is the representation of the XOR logic function. In the XOR problem there are four input points:

$$(0,0),(0,1),(1,0),(1,1)$$

The requirement is to create a classifier pattern which gives response 0 for the inputs $(0,0),(1,1)$ and 1 for the remaining.
This problem cannot be solved by a single perceptron, i.e. by a neural network without hidden layers for which the output is given by

$$y = \sum_{j=1}^{N} x_j w_j \tag{2.12}$$

Indeed, a linear neural network can only solve linear separation problem like the representation of the AND logic function.

We can show this with a figure:

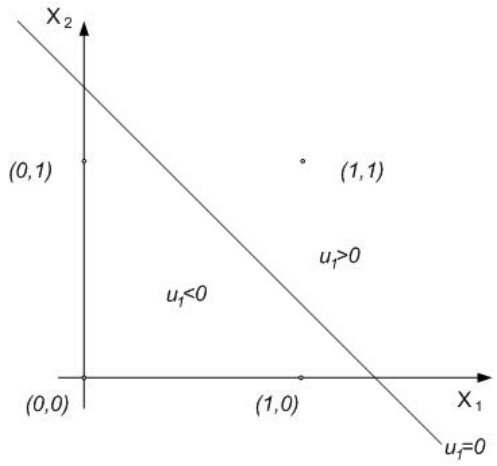


Figure 2.2: Output from linear neural network

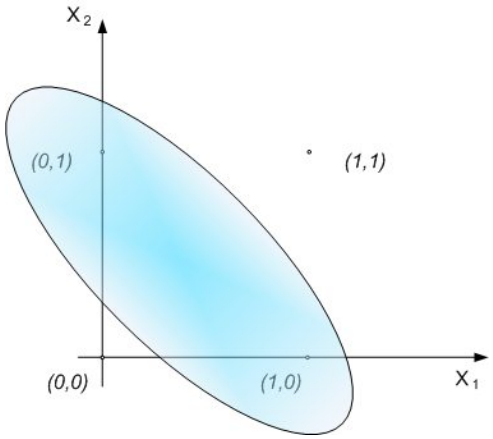


Figure 2.3: Output from non-linear neural network with hidden layer

We build explicitly the network.
Consider the two Gaussian functions:

$$\varphi_1(x) = e^{-\|x-t_1\|^2} \quad t_1 = [1,1]^T \tag{2.13}$$

$$\varphi_2(x) = e^{-\|x-t_2\|^2} \quad t_2 = [0,0]^T \tag{2.14}$$

The input patterns are mapped onto the plane $\varphi_1(x) - \varphi_2(x)$ such that the points $(0,0), (1,1)$ are linearly separable from the other even if we have not mapped the points in a high dimensional space. In other words, non-linearity exemplified by the use of Gaussian hidden function is sufficient to transform the XOR problem into a linear separable one.

From the discussion above we have learnt that a non-linear mapping is useful to transform a non-linearly separable problem in a linearly separable one. Now if we also consider that the function from the hidden layer to the output is linear, we can represent the network as the following application:

$$s(x) : \mathbb{R}^{m_1} \to \mathbb{R} \tag{2.15}$$

We may think of the map s as an hypersurface in $\Gamma \subset \mathbb{R}^{m_1+1}$. Thus Γ is a multidimensional plot of the output as a function of the input. In a more general situation the data are contaminated by noise and the surface is usually unknown. So the learning phase and the generalization phase can be viewed as the following steps [5]:

- the learning phase is the optimization of a fitting procedure for the surface Γ based on the presentation of the input and output patterns composed by the known data points.
- Generalization is synonymous with interpolation among the data points. The interpolation is made along the constrained surface generated by the previous fitting procedure as the optimum approximation to true Γ surface.

This way we are led to a multivariable interpolation problem in a high dimensional space that can be stated as follow:

Given a set of N distinct vectors (data sites)

$$\{\mathbf{x}_i \in \mathbb{R}^{m_1} | i = 1, 2, ..., N\}$$

and N real numbers $\{d_i | i = 1, 2, ..., m\}$, choose a function $s : \mathbb{R}^N \to \mathbb{R}$ which satisfies the interpolation condition

$$s(\mathbf{x}_i) = d_i \tag{2.16}$$

As we know from the previous chapter the radial basis function technique consists in choosing a function s such that:

$$s(\mathbf{x}) = \sum_{i=1}^{N} w_i \varphi(\|\mathbf{x} - \mathbf{x}_j\|) \tag{2.17}$$

where the function φ is a radial basis function.

We can see that in order to obtain the weights that define the network, we are led to solve an interpolation problem. Practically speaking, this means that we have to investigate whether the interpolation matrix is singular or not.

Chapter 3

Simultaneous approximation of function and its derivatives

In this chapter, referring to May-Duy and Tran-Cong papers (see [6, 9] for more details), we present a method to approximate simultaneously a function and its derivatives of any order, through the use of a RBF network. Then we apply this method to solve a system of ODE.

3.1 Direct and Indirect Methods

Let us take a function $f : \Omega \subset \mathbb{R}^s \to \mathbb{R}$. We want to approximate this function with positive definite radial basis functions.
The functions we are interested in are Gaussians, Hardy Multiquadrics and Hardy Inverse Multiquadrics, which from now on we indicate with Multiquadrics and Inverse Multiquadrics. Differently from the introductory chapter, in what follows, we consider a shape parameter strictly positive, such that:

- Gaussians
$$\Phi(r) = \varphi(\|\mathbf{x} - \mathbf{c}_i\|) = \exp(-\frac{r^2}{a_i^2}) \tag{3.1}$$
- Multiquadrics
$$\Phi(r) = \varphi(\|\mathbf{x} - \mathbf{c}_i\|) = \sqrt{r^2 + a_i^2} \tag{3.2}$$
- Inverse Multiquadrics
$$\Phi(r) = \varphi(\|\mathbf{x} - \mathbf{c}_i\|) = \frac{1}{\sqrt{r^2 + a_i^2}} \tag{3.3}$$

where a_i is the *width* of the radial basis function, defined as

$$a_i = \beta d_i \tag{3.4}$$

β is a factor strictly positive and d_i is the distance from the ith center $\mathbf{c}_i$ to the nearest neighbouring center.
With these information the *RBF network* can be written as:

$$s(\mathbf{x}) = \sum_{i=1}^{m} w_i \varphi(\|\mathbf{x} - \mathbf{c}_i\|). \tag{3.5}$$

In order to determine the weights by using the general least square method (this correspond to the training of the network), we should impose the following conditions:

$$f(\mathbf{x}_j) = s(\mathbf{x}_j) \quad \mathbf{x}_j \in \Omega,\ j = 1, ..., n \tag{3.6}$$

Thus, we are led to the following $n \times m$ system of linear equations:

$$\begin{pmatrix} \varphi(\|\mathbf{x}_1 - \mathbf{c}_1\|) & \varphi(\|\mathbf{x}_1 - \mathbf{c}_2\|) & \varphi(\|\mathbf{x}_1 - \mathbf{c}_3\|) & \cdots & \varphi(\|\mathbf{x}_1 - \mathbf{c}_m\|) \\ \varphi(\|\mathbf{x}_2 - \mathbf{c}_1\|) & \varphi(\|\mathbf{x}_2 - \mathbf{c}_2\|) & \varphi(\|\mathbf{x}_2 - \mathbf{c}_3\|) & \cdots & \varphi(\|\mathbf{x}_2 - \mathbf{c}_m\|) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varphi(\|\mathbf{x}_n - \mathbf{c}_1\|) & \varphi(\|\mathbf{x}_n - \mathbf{c}_2\|) & \varphi(\|\mathbf{x}_n - \mathbf{c}_3\|) & \cdots & \varphi(\|\mathbf{x}_n - \mathbf{c}_m\|) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

In what follows we indicate with $G \in \mathbb{R}^{n \times m}$ the matrix of the system and with $\mathbf{w} \in \mathbb{R}^m$ the column vector of the weights to be determined.
If we use the least square method to compute the weights w_i, we obtain the following system:

$$(G^T G)\mathbf{w} = G^T \mathbf{y} \tag{3.7}$$

Moreover, if we consider a number of centers which equals the number of data sites (i.e $m = n$) the problem becomes an interpolation problem and so we are led to the square system:

$$G\mathbf{w} = \mathbf{y}. \tag{3.8}$$

Now we can easily compute the approximation of the function through

$$f(\mathbf{x}_j) \approx \sum_{i=1}^{m} w_i \varphi(\|\mathbf{x}_j - \mathbf{c}_i\|) \qquad \forall \mathbf{x}_j \in \Omega, \quad j = 1, ..., n \tag{3.9}$$

and the approximation of the derivatives simply differentiating the radial basis function network

$$\frac{\partial^k f}{\partial x_j \cdots \partial x_l} = f_{j,..,l}(\mathbf{x}) \approx \sum_{i=1}^{m} w_i \frac{\partial^k \varphi}{\partial x_j \cdots \partial x_l}. \tag{3.10}$$

The authors called this method DRBFN, that means Direct Radial Basis Function Network. They tested the method on the following scalar valued function (i.e. $\Omega \subset \mathbb{R}$)

$$f(x) = x^3 + x + 0.5 \quad -3 \leq x \leq 2 \tag{3.11}$$

choosing the parameter $\beta = 2$.

For this function we show the approximation procedure of the function itself and of its first and second derivatives.
The Matlab script *MayDRBFNtest.m* tests the network on 250 equally spaced points using the weights computed by the Matlab function *MayDRBFNtrain.m*, which uses 50 equally spaced points as centers and as data sites; indeed the centers coincide with the data sites.
In Figure 3.1 we show the plots of the approximation by using the DRBF network, while in Table 3.1 the l_2-error.

Table 3.1: Error of DRBFN method

	Function	First Derivative	Second Derivative
G	$6.4e-01$	$2.8e+01$	$1.1e+03$
IM	$7.2e-01$	$3.0e+01$	$1.1e+03$
M	$9.2e-02$	$3.9e+00$	$1.5e+02$

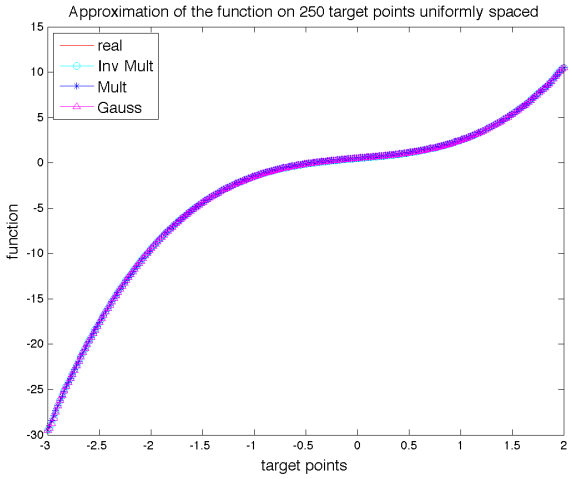
As we can see from the plots and from the error table, the approximation is not good. This is not surprising since the differentiation process is very sensitive to even a small level of noise. What we expect is that if instead of differentiating we integrate we should have a method less sensitive to noise. For this reason we can look for an *indirect method*. In fact, better results can be obtained by the following indirect method that we call IRBFN, i.e. Indirect Radial Basis Function Network. The key idea of the method is that we can simultaneously approximate a function and its derivatives up to order n, simply approximating the nth derivative with (3.9) and then computing n integrations. This means:

$$\begin{aligned} & f^{(n)}(\mathbf{x}) \approx \textstyle\sum_{i=1}^{m} w_i \Phi(\mathbf{x}) \\ & f^{(n-1)}(\mathbf{x}) = \textstyle\int f^{(n)}(\mathbf{x}) d\mathbf{x} \approx \sum_{i=1}^{m} w_i \int \Phi(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{m} w_i H^1(\mathbf{x}) + C_1 \\ & \vdots \\ & f(\mathbf{x}) = \textstyle\int f^{(1)}(\mathbf{x}) d\mathbf{x} \approx \sum_{i=1}^{m} w_i H^n(\mathbf{x}) + C_1 \mathbf{x}^{n-1} + C_2 \mathbf{x}^{n-2} + ... + C_n \end{aligned} \tag{3.12}$$

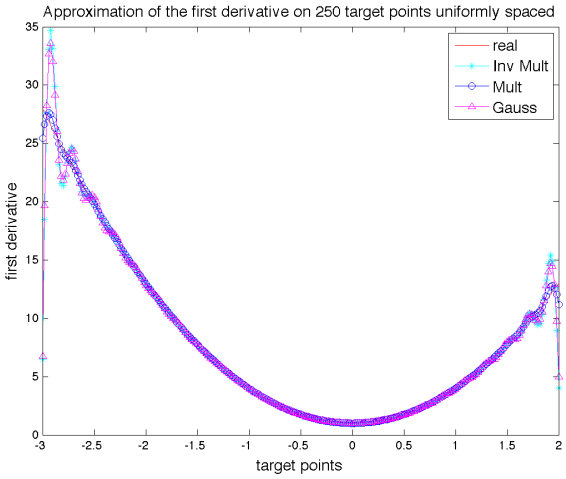
As an example, we applied this indirect method to the approximation of the function (3.11) and of its first and second derivatives.
Clearly, in this example we have to estimate the weights and two constants C_1 and C_2. This leads to a non-square system (since we have two more columns). In order to obtain a square system we may use a number of centers such that: $centers = datasites - 2$. In this way, the last two rows of the matrix G provide the equations that allow to compute the constants C_1 and C_2.

Figure 3.1: Approximation with DRBFN

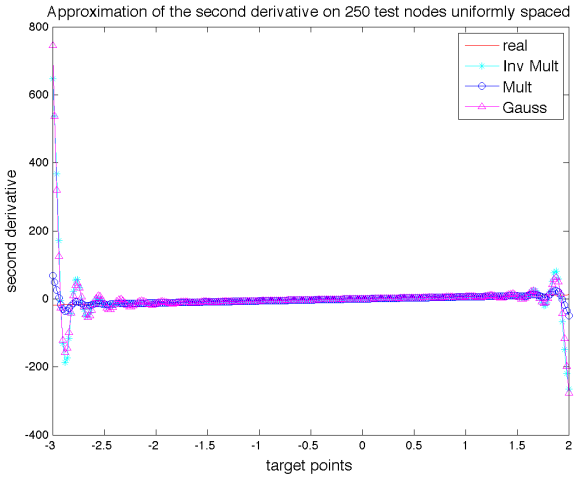
(a) Function Approximation with DRBFN



(b) First Derivative Approximation with DRBFN



(c) Second Derivative Approximation with DRBFN



The indirect method implementation is in the Matlab script *MayIRBFN2test.m* which uses the function *MayIRBFN2train.m* to compute the weights. We show the plots corresponding to the approximation obtained by the indirect method in Figure 3.2, while the error table is Table 3.2.

Table 3.2: Error of IRBFN method

	Function	First Derivative	Second Derivative
G	$2.0e-04$	$1.1e-02$	$6.2e-01$
IM	$2.0e-04$	$1.1e-02$	$6.0e-01$
M	$1.8e-05$	$9.2e-04$	$5.0e-02$

3.2 Numerical Experiments with other RBF

The following subsections describe a series of further numerical experiments we made with other RBF to better understand the functioning of the previous methods (DRBFN, IRBFN).

3.2.1 Multiquadrics

Multiquadrics are conditionally positive definite of order $k = \max(0, \lceil \beta \rceil)$. Since in our case $\beta = \frac{1}{2}$, we have $k = 1$.
As we discussed in the first chapter, in order to guarantee the non-singularity of the interpolation matrix, it is necessary to add a polynomial of degree $k-1$ to the interpolation formula (see 1.27).
However, in the specific case of generalized multiquadrics the following theorem holds (see [1], chapter 9, page 77):

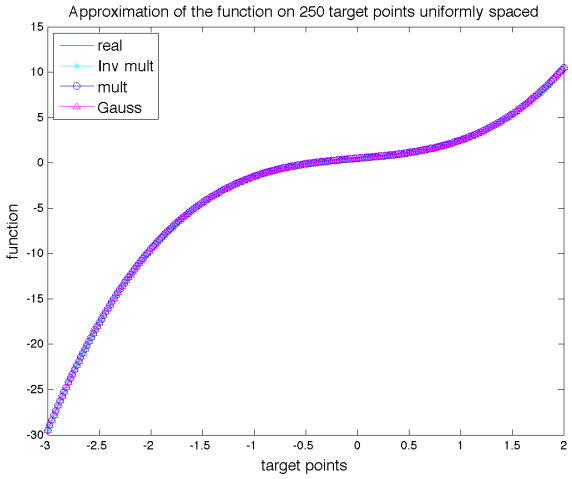
Theorem 3.1 *Suppose Φ is strictly conditionally positive definite of order one and that $\Phi(\mathbf{0}) \leq 0$. Then for any distinct points $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^s$ the matrix A with entries $A_{jk} = \Phi(\mathbf{x}_j - \mathbf{x}_k)$ has $n-1$ positive and one negative eigenvalue, and is therefore non-singular.*

What does it happen if, instead of using the previous theorem, we use the following scheme to approximate the function (3.11) and its first and second derivatives?

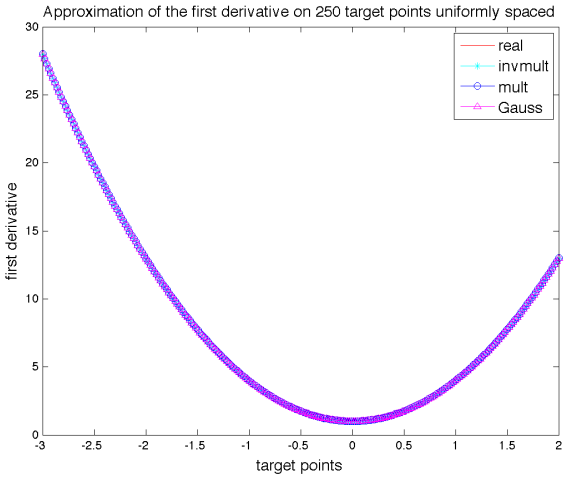
$$\begin{aligned} f''(x) &= \textstyle\sum_{i=1}^{m} w_i \Phi(x) + C_1 \\ f'(x) &= \textstyle\sum_{i=1}^{m} w_i H^1(x) + C_1 x + C_2 \\ f(x) &= \textstyle\sum_{i=1}^{m} w_i H^2(x) + C_1 x^2 + C_2 x + C_3 \end{aligned} \tag{3.13}$$

Figure 3.2: Approximation with IRBFN

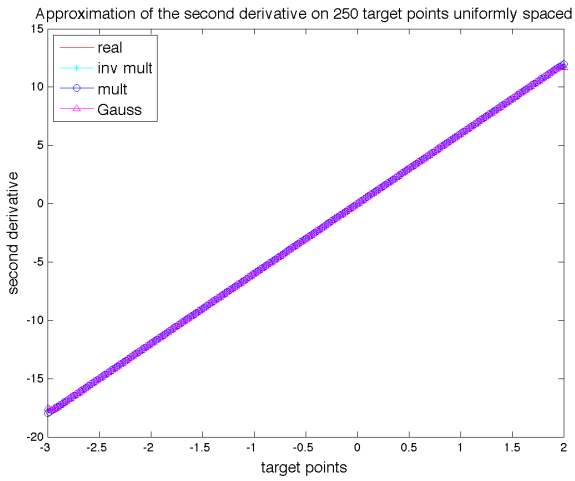
(a) Function Approximation with IRBFN



(b) First derivative Approximation with IRBFN



(c) Second derivative Approximation with IRBFN

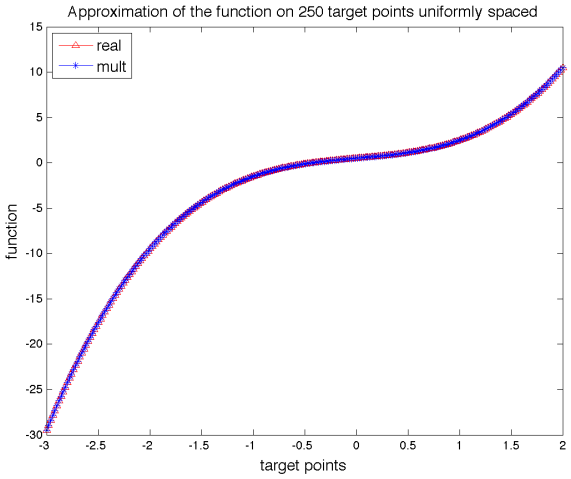


Moreover, to compute the weights we have to add the conditions:

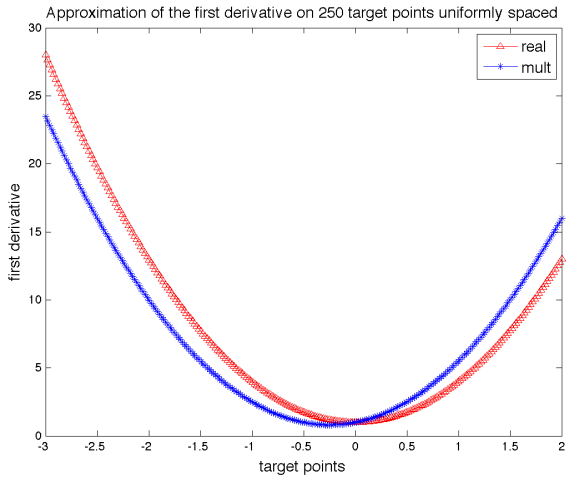
$$\sum_{i=1}^{m} w_i = 0 \tag{3.14}$$

Figure 3.3: Experiment with Multiquadrics with $C_1 \neq 0$

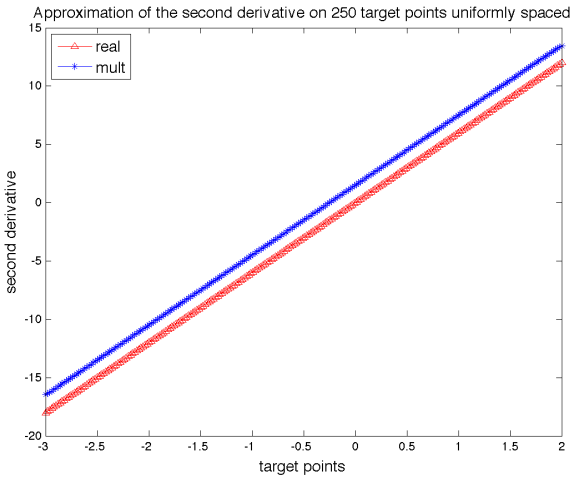
(a)



(b)



(c)



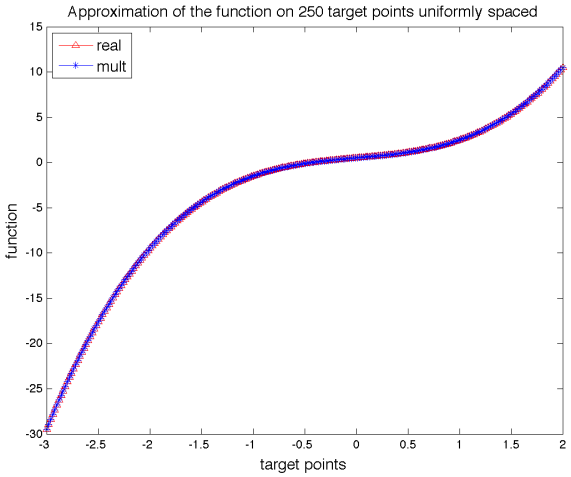
From Figure 3.4 we can see that the solution is accurate but differs from the original one for a linear polynomial, i.e. an aliasing phenomenon occurs. This is significant of the fact that the constant C_1 is superfluous, i.e. should be chosen equal to zero. Indeed, in this case, still considering the conditions on the weights, we obtain a good approximation. Table 3.3 shows the l_2-error in both cases.

Table 3.3: Error of IRBFN method

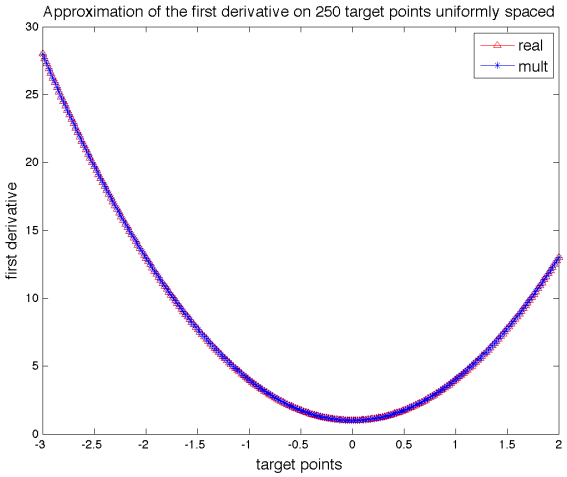
	Function	First Derivative	Second Derivative
M	$1.8e-05$	$9.2e-04$	$5.0e-02$
M with $C_1 \neq 0$	$2.0e-05$	$3.6e+01$	$2.4e+01$
M with $C_1 = 0$	$6.9e-04$	$3.0e-02$	$1.2e+00$

Figure 3.4: Experiment with Multiquadrics with $C_1 = 0$

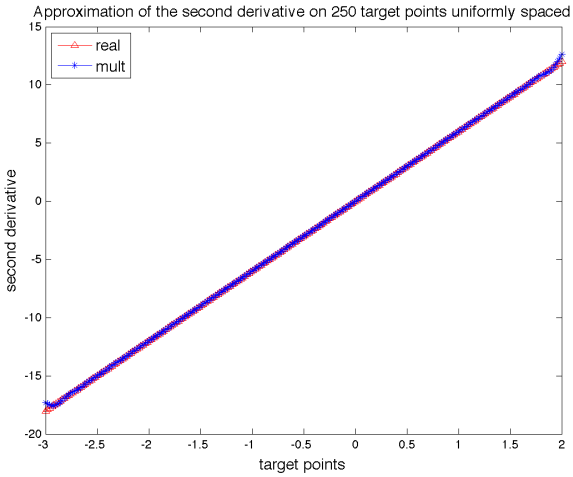
(a)



(b)



(c)



3.2.2 Thin Plate Splines

As we did in the previous section, we apply the indirect method (IRBFN) to approximate the function $y(x) = x^3 + x + 0.5$ in the interval $-3 \leq x \leq 2$ using Thin Plate Splines

$$\Phi(\mathbf{x}) = \|\mathbf{x}\|^{2\beta} \log \|\mathbf{x}\| \quad \mathbf{x} \in \mathbb{R}^s \ \beta \in \mathbb{N}$$

Choosing $\beta = 1$, the function becomes strictly conditionally positive definite of order $k = 2$. Hence Theorem 3.1 is not valid any more.
Since $k = 2$, we should use the following formulation to solve the interpolation problem.

$$\begin{aligned} f''(x) &= \textstyle\sum_{i=1}^{m} w_i \|x\|^2 \log \|x\| + C_1 x + C_2 \\ f'(x) &= \textstyle\sum_{i=1}^{m} w_i \frac{1}{9} \|x\|^3 (3 \log \|x\| - 1) + C_1 x^2 + C_2 x + C_3 \\ f(x) &= \textstyle\sum_{i=1}^{m} w_i \frac{1}{144} \|x\|^4 (12 \log \|x\| - 7) + C_1 x^3 + C_2 x^2 + C_3 x + C_4 \end{aligned} \tag{3.15}$$

To compute the weights we have to add the conditions (see section 1.2 for further explanation):

$$\begin{aligned} \textstyle\sum_{i=1}^{m} w_i &= 0 \\ \textstyle\sum_{i=1}^{m} w_i x_i &= 0 \end{aligned} \tag{3.16}$$

As we can see from Figure 3.5 the behaviour of the approximation is similar to the previous case of multiquadrics. Thus, we tried, to put the constants C_1 and C_2 equal to zero, still considering the conditions on the coefficients (3.16). Table 3.4 shows the corresponding l_2-errors.

Table 3.4: Error of IRBFN method

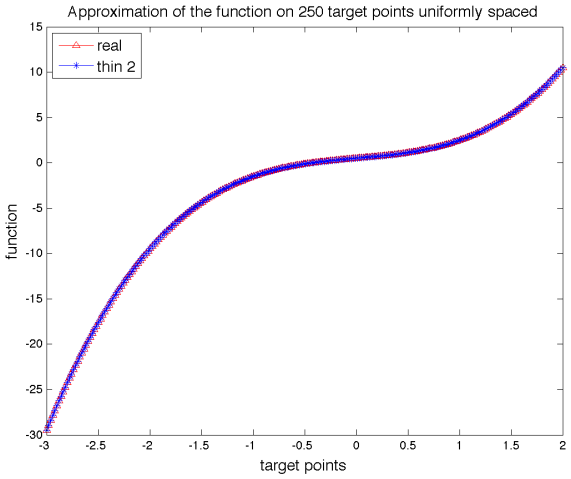
	Function	First Derivative	Second Derivative
TPS $C_1 \neq 0, C_2 \neq 0$	$6.4e-06$	$1.1e+02$	$1.2e+02$
TPS $C_1 = 0, C_2 = 0$	$3.7e-03$	$7.0e+02$	$6.0e+00$
M	$1.8e-05$	$9.2e-04$	$5.0e-02$

3.2.3 Wendland

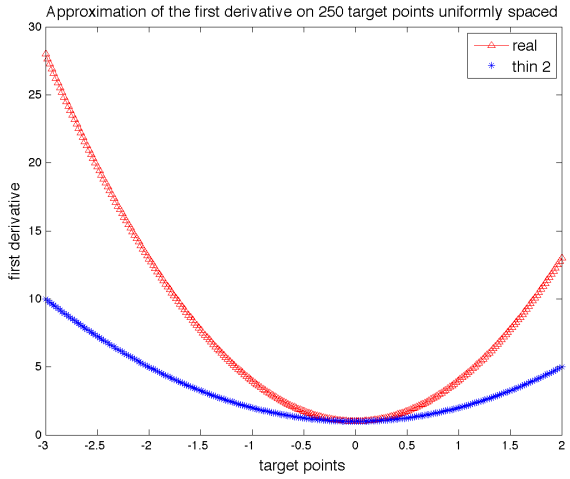
Till now we have always used globally defined radial basis functions. In this section we illustrate some results obtained using compactly supported radial basis functions. In particular we refer to Wendland functions.

Figure 3.5: Experiment with Thin Plate Splines with $C_1 \neq 0, C_2 \neq 0$

(a)



(b)



(c)

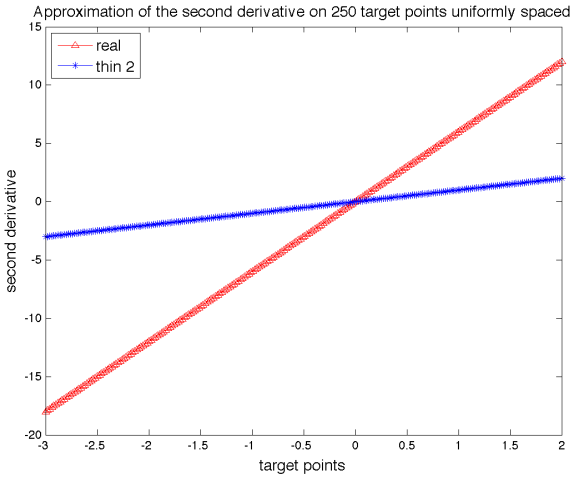
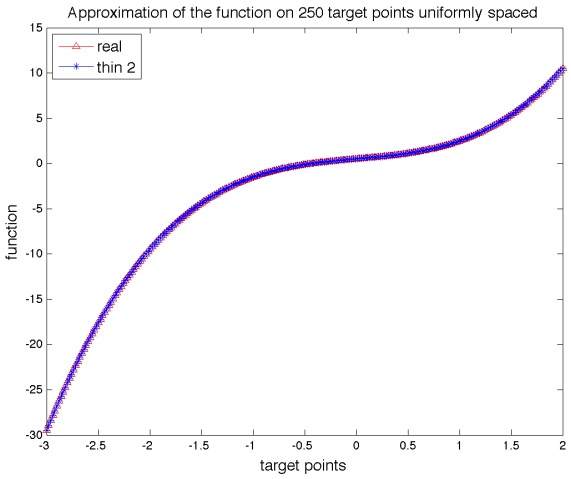
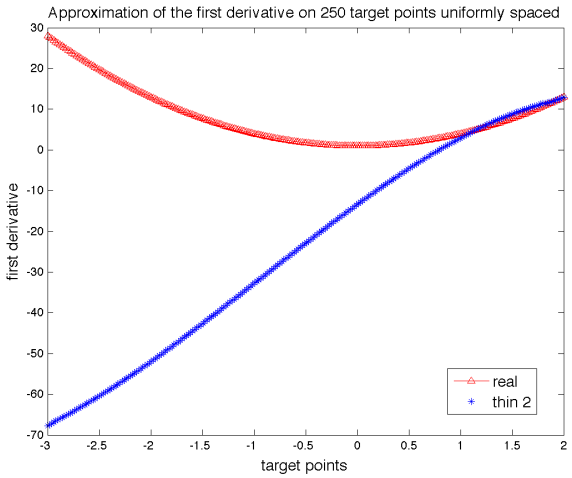


Figure 3.6: Experiment with Thin Plate Splines with $C_1 = 0, C_2 = 0$

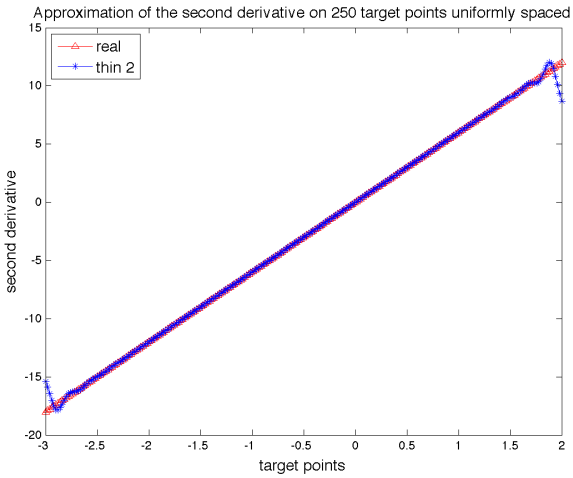
(a)



(b)



(c)



3.2.3.1 Wendland functions construction

At first we have to notice (see [1] for further explanation) that compactly supported functions Φ which are truly strictly conditionally positive definite of order $k > 0$ do not exist. Indeed, the compact support automatically ensures that Φ is strictly positive definite.
Another observation is that compactly supported radial functions can be strictly positive definite on $\mathbb{R}^s$ only for a fixed maximal s-value. Indeed, a function is strictly positive definite and radial on $\mathbb{R}^s$ if its s-variate Fourier transform is non-negative.
After this two considerations, we can introduce an integral operator and its inverse, a differential operator, which can facilitate the construction of compactly supported radial functions.

Definition 3.1 *Let φ be such that $t \to t\varphi(t) \in L_1[0,\infty)$. Then we define the integral function operator I via*

$$(I\varphi)(r) = \int_r^\infty t\varphi(t)dt \quad r \geq 0 \tag{3.17}$$

For even $\varphi \in C^2(\mathbb{R})$ we define the differential operator D via

$$(D\varphi)(r) = -\frac{1}{r}\varphi'(r) \quad r \geq 0 \tag{3.18}$$

These operators allow us to construct new strictly positive definite radial functions from given ones by a "dimension walk" technique that steps through multivariate Euclidean space in even increments.
One of this family of compactly supported functions are the Wendland functions. Wendland starts with the truncated power function $\varphi_l(r) = (1-r)_+^l$ and then walks through dimensions by repeatedly applying the operator I.

Definition 3.2 *With $\varphi_l(r) = (1-r)_+^l$ we define*

$$\varphi_{s,k} = I^k \varphi_{\lfloor s/2 \rfloor + k + 1}. \tag{3.19}$$

It turns out that the functions $\varphi_{s,k}$ are all supported on $[0,1]$ and have a polynomial representation there. Indeed,

Theorem 3.2 *The functions $\varphi_{s,k}$ are strictly positive definite and radial on $\mathbb{R}^s$ and are of the form*

$$\varphi_{s,k} = \begin{cases} p_{s,k}(r), & r \in [0,1] \\ 0, & r > 1 \end{cases} \tag{3.20}$$

with a univariate polynomial $p_{s,k}$ of degree $\lfloor s/2 \rfloor + 3k + 1$. Moreover, $\varphi_{s,k} \in C^{2k}(\mathbb{R}^s)$ are unique up to a constant factor, and the polynomial degree is minimal for given space dimension s and smoothness $2k$.

Some examples:

$$\varphi_{3,1} = \frac{1}{12}(1-ar)_+^4(4ar+1)$$

$$\varphi_{3,2} = \frac{1}{72}(1-ar)_+^6(35a^2r^2+18ar+3) \tag{3.21}$$

$$\varphi_{3,3} = \frac{3}{22}(1-ar)_+^8(32a^3r^3+25a^2r^2+8ar+1)$$

3.2.3.2 Numerical experiments with Wendland functions

As we did in the previous sections we approximate the polynomial $f(x) = x^3 + x + 0.5$ in the interval $[-3, 2]$, using 50 equally spaced points as centers and 250 equally spaced points as target points.
For our purposes, we use the Wendland $\varphi_{3,1}$ which is C^2 in the origin, and we differentiate it two times in order to approximate the first and the second derivative. So, we have:

$$\begin{aligned}
&\varphi_{3,1}(r) = (1-ar)_+^4(4ar+1)\\
&\tfrac{d}{dr}\varphi_{3,1}(r) = -20a^2r(1-ar)_+^3\\
&\tfrac{d^2}{dr^2}\varphi_{3,1}(r) = 20a^2(1-ar)_+^2(4ar-1),
\end{aligned}$$

We also use the Wendland $\varphi_{3,2}$ which has order C^4 at the origin:

$$\begin{aligned}
&\varphi_{3,2}(r) = (1-ar)_+^6(35a^2r^2+18ar+3)\\
&\tfrac{d}{dr}\varphi_{3,2}(r) = -56a^2r(5ar+1)(1-ar)_+^5\\
&\tfrac{d^2}{dr^2}\varphi_{3,2}(r) = 56a^2(1-ar)_+^4(35a^2r^2-4ar-1),
\end{aligned}$$

and Wendland $\varphi_{3,3}$ which has order C^6 at the origin:

$$\begin{aligned}
&\varphi_{3,3}(r) = (1-ar)_+^8(32a^3r^3+25a^2r^2+8ar+1)\\
&\tfrac{d}{dr}\varphi_{3,3}(r) = -22a^2r(16a^2r^2+7ar+1)(1-ar)_+^7\\
&\tfrac{d^2}{dr^2}\varphi_{3,3}(r) = 22a^2(1-ar)_+^6(160a^3r^3+15a^2r^2-6ar-1).
\end{aligned}$$

.

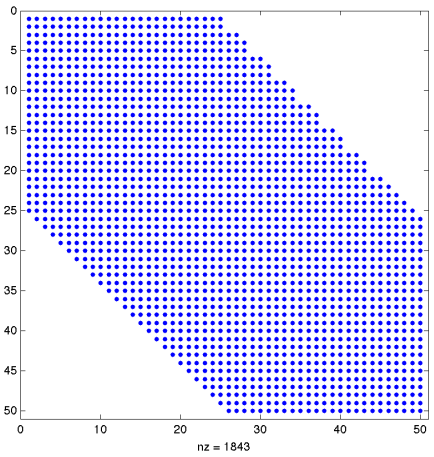
The implementations are in the Matlab scripts *training.m* and *testnetwork.m*. We have to notice that we use a different shape parameter for the Multiquadrics and for the Wendland functions, i.e. in both cases we have $d_i = 0.102$, but $\beta = 2$ for Multiquadrics and $\beta = 4$ for the Wendland functions.
What is more, to build the matrix for the Wendland we use the specific Matlab function *DistanceMatrixCSRBF_new.m* which gives a sparse matrix. Moreover, the greater the width parameter, the sparser the matrix.
In Figure 3.7 we display the plots while in Table 3.5 the l_2-errors:

Table 3.5: Error of DRBFN method

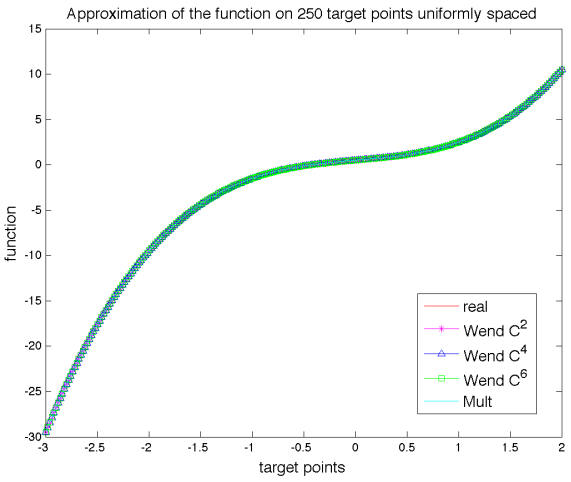
	Function	First Derivative	Second Derivative
M	$9.2e-02$	$3.9e+00$	$1.5e+02$
$\varphi_{3,1}$	$1.5e-01$	$1.8e+02$	$2.6e+02$
$\varphi_{3,2}$	$4.6e-02$	$4.5e+02$	$8.1e+01$
$\varphi_{3,3}$	$7.7e-03$	$2.7e+03$	$1.5e+01$

Figure 3.7: Direct Method using Wendland function as basis

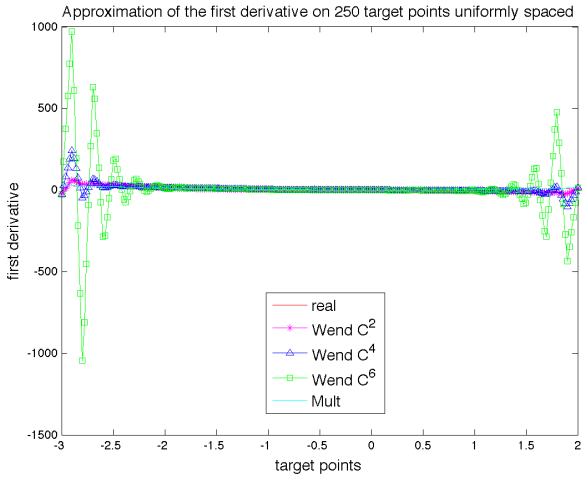
(a) Matrix Sparsity



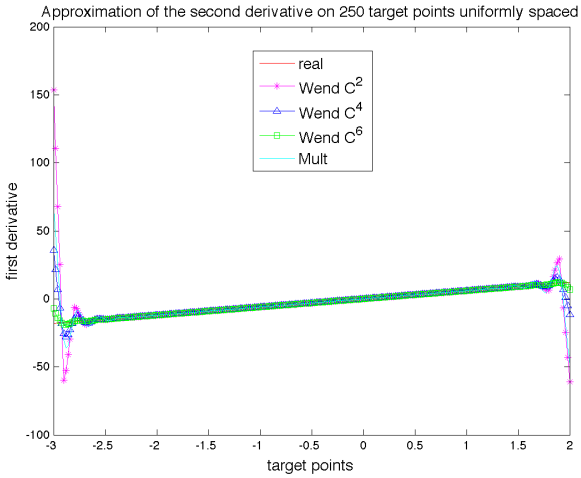
(b) Function Approximation



(c) First Derivative Approximation



(d) Second Derivative Approximation



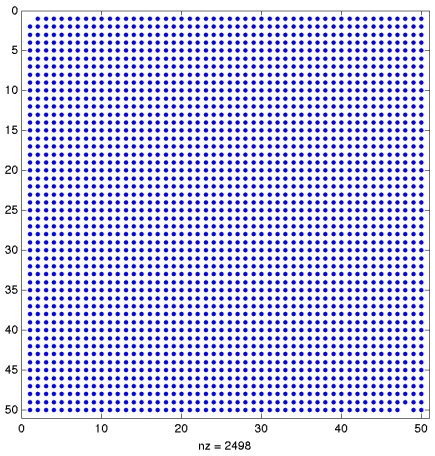
We also try to apply the indirect method (IRBFN). We integrate $\varphi_{3,1}$, $\varphi_{3,2}$, $\varphi_{3,3}$ and $\varphi_{3,4}$. Table 3.6 shows the l_2-error.

Table 3.6: Error of IRBFN method

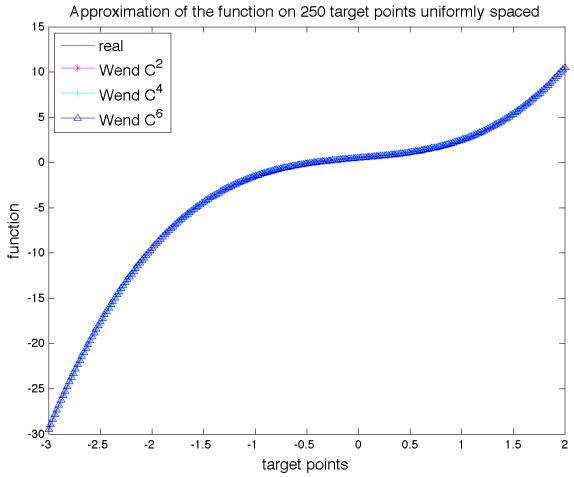
	Function	First Derivative	Second Derivative
M	$1.8e-05$	$9.2e-04$	$5.0e-02$
$\varphi_{3,1}$	$1.9e-02$	$1.5e+04$	$7.8e-01$
$\varphi_{3,2}$	$3.7e-03$	$3.4e+04$	$1.6e-01$
$\varphi_{3,3}$	$1.1e-03$	$9.8e+04$	$5.0e-02$

Figure 3.8: Indirect Method using Wendland function as basis

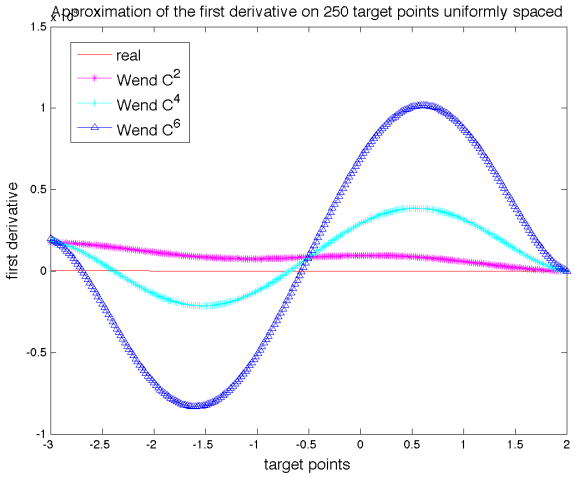
(a) Matrix sparsity



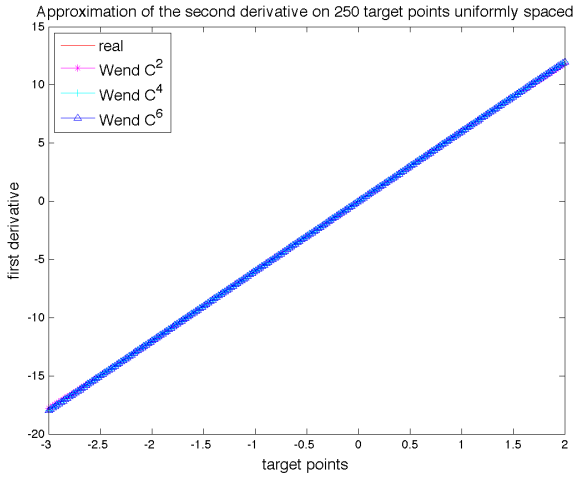
(b) Function Approximation



(c) First Derivative Approximation

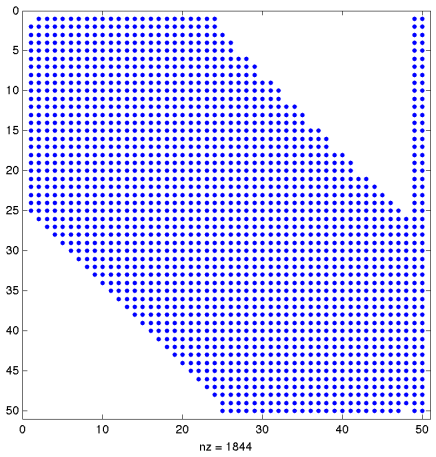


(d) Second Derivative Approximation



In this case it seems, from a *trial and error* approach, that the best choice for the shape

parameter is $a_i = 0.2 \cdot 0.102$. This choice leads, as we can see from Figure 3.8 to a dense matrix.
This behaviour, can be explained by observing that in the indirect method, it is necessary to add two more columns to the matrix which computes the weights, with the aim to estimate the two integration constants C_1 and C_2 (see 3.12).
The columns' addition breaks the sparse structure as we show in the following figure:



Finally we observe that the worst approximation is obtained in evaluating the first derivative, both for Thin Plate Splines and Wendland functions.
This is due to the fact that the integral and differential operators, which define Wendland functions, allow a "dimension walk" that steps through multivariate Euclidean space in even increments. Thus, it is only through the double application of the differential operator that we obtain a function that is still in the Wendland native space.
The same is true also for Thin Plate Splines which are defined only for parameters $\beta \in \mathbb{N}$, i.e.

$$\Phi(\mathbf{x}) = \|\mathbf{x}\|^{2\beta} \log \|\mathbf{x}\| \quad \mathbf{x} \in \mathbb{R}^s \ \beta \in \mathbb{N}.$$

This means that we can always obtain good approximation for even derivatives but not for the odd ones.

3.2.4 Considerations on Native spaces

In section 1.3 we have listed the native spaces for some radial basis functions. In particular we saw that Wendland's compactly supported functions have native spaces

$$N_{\Phi_{s,k}}(\mathbb{R}^s) = W_2^{s/2+k+1/2}(\mathbb{R}^s) \tag{3.22}$$

while Thin Plate Splines have the so-called *Beppo-Levi space* of order k:

$$BL_k = \{f \in C(\mathbb{R}^s) : D^\alpha f \in L_2(\mathbb{R}^s) \text{ for all } |\alpha| = k, \alpha \in \mathbb{N}^s\}. \tag{3.23}$$

In order to obtain the error estimates in (3.26) and (3.27) we state some preliminaries results (proofs are in [7, 8]).

Theorem 3.3 *Suppose that $\Omega \subset \mathbb{R}^s$ is bounded and satisfies an interior cone condition. Let $l \in \mathbb{N}_0$ and $\alpha \in \mathbb{N}_0^s$ with $|\alpha| \leq l$. Then there exist constants $h_0, c_1^{(\alpha)}, c_2^{(\alpha)} > 0$ such that for all $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\} \subset \Omega$ with $h_{X,\Omega} \leq h_0$ and every $\mathbf{x} \in \Omega$ there exist numbers $\tilde{u}_1^\alpha(\mathbf{x}), ..., \tilde{u}_N^\alpha(\mathbf{x})$ with*

1. $\sum_{j=1}^N \tilde{u}_j^\alpha(\mathbf{x}) p(\mathbf{x}_j) = D^\alpha p(\mathbf{x})$ *for all* $p \in \pi_l(\mathbb{R}^s)$,
2. $\sum_{j=1}^N |\tilde{u}_j^\alpha(\mathbf{x})| \leq c_1^{(\alpha)} h_{X,\Omega}^{-|\alpha|}$,
3. $\tilde{u}_j^\alpha(\mathbf{x}) = 0$, *if* $\|\mathbf{x} - \mathbf{x}_j\|_2 > c_2^{(\alpha)} h_{X,\Omega}$.

The next result deals with (conditionally) positive definite functions.

Theorem 3.4 *Suppose that $\Phi \in C^k(\mathbb{R}^s)$ is conditionally positive definite of order m. Suppose further that $\Omega \subset \mathbb{R}^s$ is bounded and satisfies an interior cone condition. Fix $l \geq m-1$. For $\alpha \in \mathbb{N}_0^s$ with $|\alpha| \leq k/2$ and $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\} \subset \Omega$ satisfying $h_{X,\Omega} \leq h_0$ the power function can be bounded:*

$$\begin{aligned} [P_{\Phi,X}^\alpha(\mathbf{x})]^2 &\leq |D^{2\alpha}\Phi(0) - D^{2\alpha}p(0)| \\ &+ 2c_1^{(\alpha)} h_{X,\Omega}^{-|\alpha|} \|D^\alpha \Phi - D^\alpha p\|_{L_\infty(B(0, c_2^{(\alpha)} h_{X,\Omega}))} \\ &+ [c_1^{(\alpha)}]^2 h_{X,\Omega}^{-2|\alpha|} \|\Phi - p\|_{L_\infty(B(0, 2c_2^{(\alpha)} h_{X,\Omega}))} \end{aligned} \tag{3.24}$$

where p is an arbitrary polynomial from $\pi_l(\mathbb{R}^s)$ and the constants $h_0, c_1^{(\alpha)}, c_2^{(\alpha)}$ come from Theorem 3.3.

Now we can state a generic error estimate

Theorem 3.5 *Suppose that $\Phi \in C_\nu^k(\mathbb{R}^s)$ is conditionally positive definite of order m. Suppose further that $\Omega \subset \mathbb{R}^s$ is bounded and satisfies an interior cone condition. For $\alpha \in \mathbb{N}_0^s$ with $|\alpha| \leq k/2$ and $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\} \subset \Omega$ satisfying $h_{X,\Omega} \leq h_0$ we have the error bound*

$$\|D^\alpha f(\mathbf{x}) - D^\alpha P_f(\mathbf{x})\|_{L_\infty}(\Omega) \leq C h_{X,\Omega}^{(k+\nu)/2-|\alpha|} \|f\|_{N_\Phi(\Omega)} \tag{3.25}$$

Now we can apply the general result of Theorem 3.5 to Wendland and Thin Plate Splines functions.

- **Wendland Function.**

$$|D^\alpha f(\mathbf{x}) - D^\alpha P_f(\mathbf{x})| \leq C h_{X,\Omega}^{k+\frac{1}{2}-|\alpha|} \|f\|_{N_\Phi(\Omega)} \tag{3.26}$$

for every $|\alpha| \leq k$, $h_{X,\Omega}$ sufficiently small.

Remark. We have to notice that in the case of $\varphi_{3,1}$ we can use this estimate only for the function and first derivative, since $|\alpha| \leq k$.

- **Thin Plate Splines.**

$$|D^\alpha f(\mathbf{x}) - D^\alpha P_f(\mathbf{x})| \leq C h_{X,\Omega}^{k-|\alpha|} |f|_{N_\Phi(\Omega)} \tag{3.27}$$

  for every $|\alpha| \leq k-1$, $h_{X,\Omega}$ sufficiently small.

In order to obtain the error estimates we should also compute the Sobolev norm and Beppo-Levi semi-norm of the function f.

$$\begin{aligned} \|f\|_{W_p^m(\Omega)} &= \left(\sum_{|\alpha|\leq m} \|D^\alpha f\|_{L_p(\Omega)}^p\right)^{1/p} \\ |f|_{BL_k} &= \left(\sum_{|\alpha|=m} \frac{m!}{\alpha!} \|D^\alpha f\|_{L_p(\Omega)}^p\right)^{1/p} \end{aligned} \tag{3.28}$$

A function that belongs to these spaces is the gaussian $e^{(-x^2)}$. For this function we compute an estimate of the constant C in (3.26), by the following computation:

$$C = \frac{|D^\alpha f(\mathbf{x}) - D^\alpha P_f(\mathbf{x})|}{h_{X,\Omega}^{k+\frac{1}{2}-|\alpha|} \|f\|_{N_\Phi}} \tag{3.29}$$

By considering the maximum of C we obtain a first estimate of the error. We also try to take the mean value of vector C, but this leads to underestimate the error.
We display the error plots for $\varphi_{3,1}$, $\varphi_{3,2}$ and $\varphi_{3,3}$ in Figure 3.9 and 3.11, where in black is the error estimate when C assumes its maximum value.

In the case of Thin Plate Splines, since we are considering $k = 1$ the (3.27) is valid only for the function approximation. For the estimate of the constant C we proceed as in the previous case, by considering the following formula:

$$C = \frac{|D^\alpha f(\mathbf{x}) - D^\alpha P_f(\mathbf{x})|}{h_{X,\Omega}^{k-|\alpha|} |f|_{N_\Phi}} \tag{3.30}$$

In Figure 3.11 we display the error trend for the function approximation by using Thin Plate Splines. The black points are the error estimate obtained by taking the maximum value assumed by C.

On the other hand, it is quite easy to see that the polynomials are not in these two spaces. So the previous results are not valid. However, the following theorem holds:

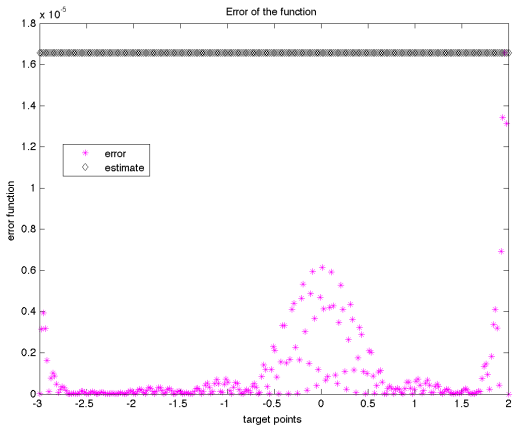
Theorem 3.6 *Let k and n be integers with $0 \leq n < k \leq \tau$ and $k > s/2$, and let $f \in C^k(\bar{\Omega})$. Also suppose that $X = \{\mathbf{x}_1, ..., \mathbf{x}_N\} \subset \Omega$ satisfies $diam(X) \leq 1$ with sufficiently small fill distance $h_{X,\Omega}$. Then for any $1 \leq q \leq \infty$ we have*

$$|f - P_f|_{W_q^n(\Omega)} \leq c \rho_X^{\tau-k} h_{X,\Omega}^{k-n-s(1/2-1/q)_+} \|f\|_{C^k(\bar{\Omega})}, \tag{3.31}$$

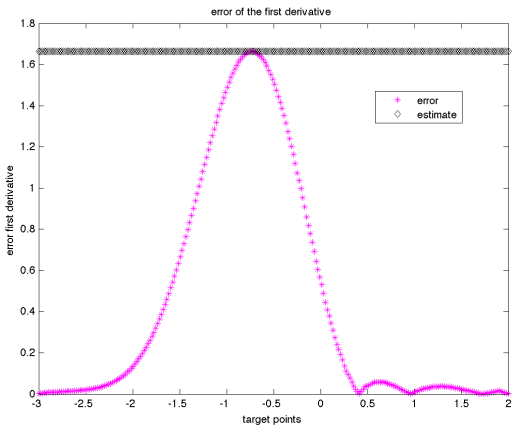
where $\rho_X = \frac{h}{q_X}$ is the mesh ratio for X and q_X is the separation distance $\frac{1}{2}\min_{i\neq j} \|x_i - x_j\|_2$.

Figure 3.9: Error trend using Wendland function as basis

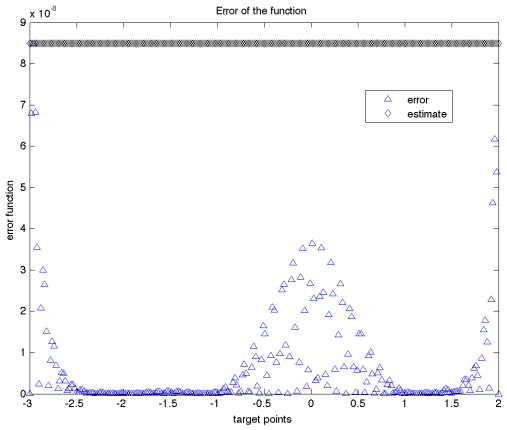
(a) error function with $\varphi_{3,1}$



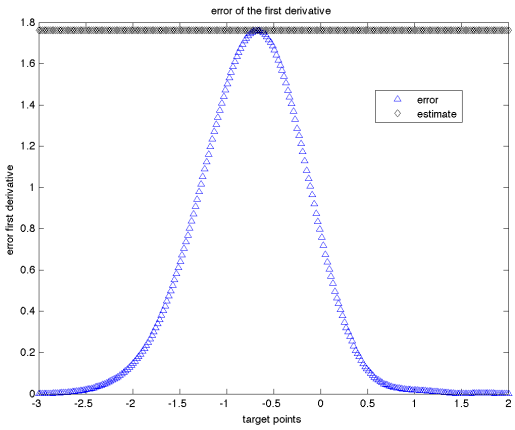
(b) Error first derivative with $\varphi_{3,1}$



(c) Error function with $\varphi_{3,2}$



(d) Error first derivative with $\varphi_{3,2}$



(e) Error second derivative with $\varphi_{3,2}$

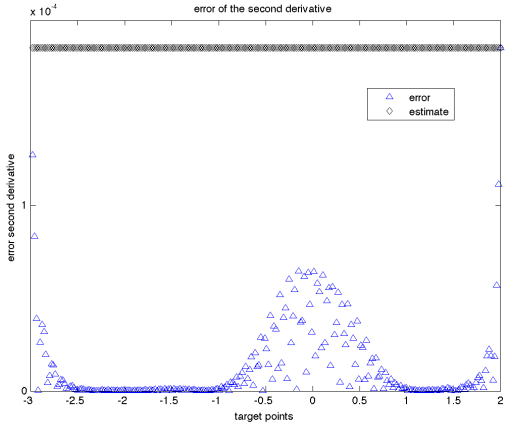
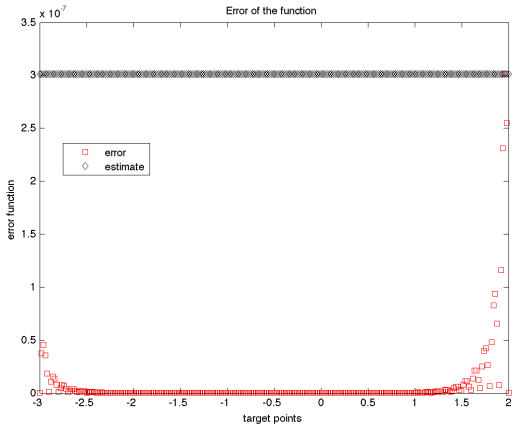
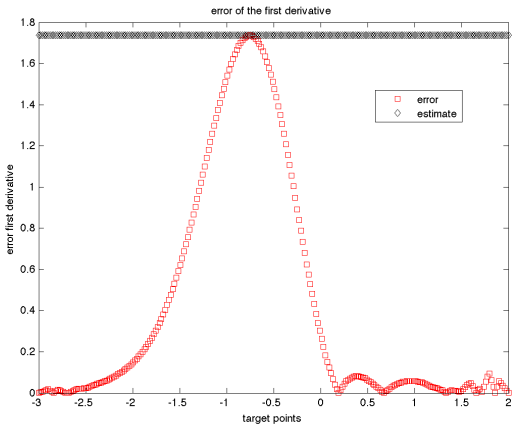


Figure 3.10: Error trend using Wendland function as basis

(a) Error function with $\varphi_{3,3}$



(b) Error first derivative with $\varphi_{3,3}$



(c) Error second derivative with $\varphi_{3,3}$

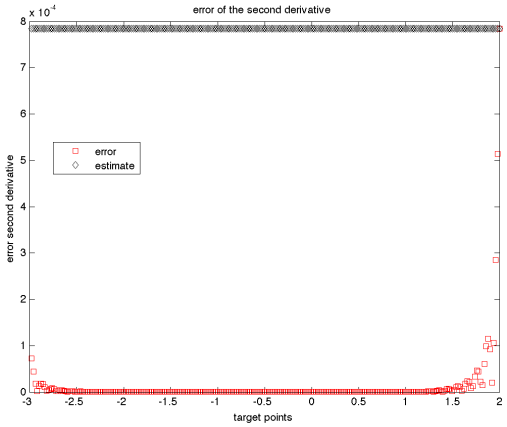
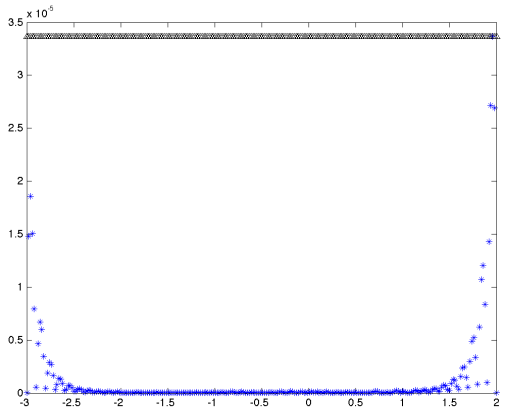


Figure 3.11: Error trend using Thin Plate Splines function as basis



In the case of Thin Plate Splines, choosing $\tau = 2\beta$ and $k = \tau, n = 0, q = \infty$ we arrive at the bound:

$$|f - P_f|_{L_\infty} \leq c h_{X,\Omega}^{2\beta - s/2} \|f\|_{C^{2\beta}}(\bar{\Omega}). \tag{3.32}$$

As we did for the function e^{-x^2} we can estimate the constant c as:

$$c = \frac{|f - P_f|_{L_\infty}}{h_{X,\Omega}^{2\beta - s/2} \|f\|_{C^{2\beta}}(\bar{\Omega})}. \tag{3.33}$$

Thus we obtain:

Table 3.7: Constant estimate for Thin Plate Splines

TPS Constant	$1.8e-03$

For Wendland functions we can choose $\tau = k$ (where k is the order of the Wendland function in the notation $\varphi_{s,k}$) and $n = 0, q = \infty$. In this way we are led to:

$$|f - P_f|_{L_\infty} \leq c h_{X,\Omega}^{k - s/2} \|f\|_{C^k(\bar{\Omega})}. \tag{3.34}$$

As for Thin Plate Splines we estimate the constant c:

Table 3.8: Constant estimate for Wendland

$\varphi_{3,1}$ constant	$6.7e-03$
$\varphi_{3,2}$ constant	$2.9e-02$
$\varphi_{3,3}$ constant	$8.9e-02$

3.2.5 Network existence and localization properties

In this section, we first state an existence result for the RBF network for simultaneous approximation of a function and its derivatives under some conditions on the kernels and the function to be approximated (see [12]). Then, we discuss localization properties for different kernels.
Let $\mathbb{Z}^s$ denote the set of all multi-integers in $\mathbb{R}^s$. For $\mathbf{k} = (k_1, ..., k_s) \in \mathbb{Z}^s$, let $|k| = \sum_{j=1}^{s} |k_j|$ and for $\mathbf{m} = (m_1, ..., m_s) \in \mathbb{Z}^s$ we say that $\mathbf{k} \leq \mathbf{m}$, if $k_j \leq m_j$ for all j, $1 \leq j \leq s$.
We denote with $D^{\mathbf{m}} f$ the $\mathbf{m}$th order partial derivative of a multivariate function f. Clearly in the univariate case $D^{\mathbf{m}} f = f^{(m)}$.
Then, we denote with $\bar{C}^{(\mathbf{m}_1,...,\mathbf{m}_q)}(K) = \bar{C}^{\mathbf{m}_1}(K) \cap ... \cap \bar{C}^{\mathbf{m}_q}(K)$ the set of all function f such that for any $k \in J(\mathbf{m}_1, ..., \mathbf{m}_q)$, $D^k f$ is continuous on an open set containing the compact K, where $\mathbf{m}_l \in \mathbb{Z}_+^s$ and $J(\mathbf{m}_1, ..., \mathbf{m}_q)$ is the set of all $\mathbf{k} \in \mathbb{N}^s$ such that $\mathbf{k} \leq \mathbf{m}_l$ for some l, $1 \leq l \leq q$.
Finally, let $s(\mathbf{x}) = \sum_{j=1}^{m} c_j \varphi(\lambda_j \|\mathbf{x} - \mathbf{t}_j\|)$, be the network for the simultaneous approximation, where $c_j \in \mathbb{R}$, $\lambda_j > 0$, $\mathbf{t}_j \in \mathbb{R}^s$ and $\varphi(\|\cdot\|)$ is the radial basis function.
The following theorem holds:

Theorem 3.7 *Suppose that a univariate function φ is analytic in $(-r, r)$, where $r > 0$, and for some integer $p \geq 1$, $\varphi^{(pj)}(0) \neq 0$ for $j \geq 0$. Then for any $f \in \bar{C}^{(\mathbf{m}_1,...,\mathbf{m}_q)}([-\pi, \pi]^s)$, and any $\epsilon > 0$, there is a RBF neural network*

$$s(\mathbf{x}) = \sum_{j=1}^{m} c_j \varphi(\lambda_j \|\mathbf{x} - \mathbf{t}_j\|) \tag{3.35}$$

where $c_j \in \mathbb{R}$, $\mathbf{t}_j \in [-2\pi, 2\pi]^s$, and $0 < \lambda_j < r/(9\pi^2 s)$, for $1 \leq j \leq m$, such that

$$\|D^{\mathbf{k}} f - D^{\mathbf{k}} s\|_{L_\infty([-\pi,\pi]^s)} < \epsilon \tag{3.36}$$

for all $\mathbf{k} \in J(\mathbf{m}_1, ..., \mathbf{m}_q)$.

We first remark that most RBFs studied in literature, such as Gaussians, Multiquadrics and Inverse Multiquadrics satisfy the assumptions of Theorem 3.7.
Moreover, by considering the convolution operator and the introduction of mollifiers, we may extend the theorem to every compact set:

Corollary 3.8 *Let φ be assumed as in Theorem 3.7. Suppose that K is a compact subset of $\mathbb{R}^s$, and $f \in \bar{C}^{(\mathbf{m}_1,...,\mathbf{m}_q)}(K)$. Then for any $\epsilon > 0$, there is a network s in the form of equation (3.35) for some suitable $c_j \in \mathbb{R}$, $\lambda_j \in \mathbb{R}$, and $\mathbf{t}_j \in \mathbb{R}^s$, where $1 \leq j \leq m$, such that*

$$\|D^{\mathbf{k}} f - D^{\mathbf{k}} s\|_{L_\infty(K)} < \epsilon \tag{3.37}$$

for any $\mathbf{k} \in J(\mathbf{m}_1, ..., \mathbf{m}_q)$.

Finally, we observe that the density results can also be established for the L_p norm, i.e. even if the function f is defined in a Sobolev space.
Clearly, the RBF network defined in the first section of this chapter, is a particular case of the previous, since can be obtained by choosing $\lambda_j = 1$ and $\mathbf{x} = \mathbf{t}_j$.
Referring to [18, 14], we comment the numerical results of the previous section, in which we have seen that better approximations are obtained by using Multiquadrics.
Consider the following quasi-interpolation formula:

$$s(\mathbf{x}) = \sum_{\mathbf{y}\in\mathbb{Z}^n} f(\mathbf{y})\psi(\mathbf{x}-\mathbf{y}) \quad \mathbf{x}\in\mathbb{R}^n \tag{3.38}$$

where $f : D \subset \mathbb{R}^n \to \mathbb{R}$ is the function to be approximated by s and $\mathbb{Z}^n$ is the set of vectors in $\mathbb{R}^n$ whose components are integers and ψ is of the form:

$$\psi(\mathbf{x}) = \sum_{k=1}^{m} \mu_k \varphi(\|\mathbf{x}-\mathbf{x}_k\|_2), \quad \mathbf{x}\in\mathbb{R}^n \tag{3.39}$$

We showed that radial basis function approximations can have good localization properties even if $\varphi(r) \to \infty$ as $r \to \infty$.
Indeed, it seems that such radial functions have advantages over those ones that tend to zero as $r \to \infty$.
Here *localization* means that, if $D \subset \mathbb{R}^n$ is the domain of the approximation, if $s(\cdot)$ is calculated from $f(\cdot)$, and if D_0 is any part of D, then the contribution to $s(\mathbf{x})$ from $\{f(\mathbf{y}) : \mathbf{y} \in D_0\}$ is small for values of $\mathbf{x}$ in D that are far from D_0.
Such localization properties are highly important to the success of general approximation methods.
In the one-dimensional case is quite easy to show these properties. Indeed, if we let $\varphi(r) = r$, and if we compute the coefficients $\{\mu_k : k = 1, 2, ..., m\}$ of expression (3.39) so that they satisfy conditions (3.1), then $s(\cdot)$ is the piecewise linear interpolant and we can express it in the form:

$$s(x) = \sum_{k=1}^{m} f(x_k)\psi_k(x) \quad x_1 \le x \le x_m \tag{3.40}$$

where each $\psi_i(\cdot)$ is independent of $f(\cdot)$.
For example $\psi_2(\cdot)$ is the *hat function*

$$\psi_2(x) = \frac{(x_3-x_2)|x-x_1| - (x_3-x_1)|x-x_2| + (x_2-x_1)|x-x_3|}{2(x_2-x_1)(x_3-x_2)} \; x_1 \le x \le x_m \tag{3.41}$$

In general, we have to search in the space spanned the functions $\{\varphi(\|\cdot - x_i\|_2) : i = 1, 2, ..., m\}$ if we wish to discover whether a particular $\varphi(\cdot)$ allows good localization properties.

In order to have an absolutely convergent sum for smooth $\psi(\cdot)$ and any bounded function $f(\cdot)$, we need the following *minimum localization conditions*:

$$\begin{aligned} \int_{\mathbb{R}^n} |\psi(\mathbf{x})| d\mathbf{x} < \infty \\ \int_{\mathbb{R}^n} \psi(\mathbf{x}) d\mathbf{x} = 1 \end{aligned} \tag{3.42}$$

Whether or not they can be satisfied when ψ has the form (3.39), depends on the radial function φ and on the dimension n.
In particular, in the case of multiquadrics and in the case $n = 1$, conditions on the coefficients μ_k, in formula (3.39), exist such that the minimum localization conditions are satisfied for each choice of the shape parameter.
Indeed, the conditions on μ_k are independent of the shape parameter.
For what concern the case of radial functions such that $\varphi(r) \to 0$ as $r \to \infty$, as inverse multiquadrics, in [14] it has been shown that, if we require that $\int_{-\infty}^{\infty} |\psi(x)| dx < \infty$ then we have $\int_{-\infty}^{\infty} \psi(x) dx = 0$.
This is undesirable, since it shows that the quasi-interpolant to a constant function has zero integral if ψ is absolutely integrable. Thus, the approximation is no more accurate than $s \equiv 0$ if we try to obtain good localization properties for the cardinal functions.
Finally, the thin plate splines are unsuitable for quasi interpolation when $n = 1$.
We consider also a more recent work on the localization properties of radial basis functions by Fornberg ([15]). Here a different approach is used, based on studying the behaviour of the resulting RBF expansion coefficients, i.e. μ_k in (3.39), for increasing $|k|$. It has been shown that the leading order behaviour of the expansion coefficients for small $|k|$ decays exponentially. However, some RBF can exhibit two different decays regimes for increasing k. In any case, the localization properties are determined by this exponential decay.

Chapter 4

Solving high order differential equations using RBF networks

In Chapter 3 we have studied a method to approximate a function together with its derivatives. In [9], it has been shown that it is possible to use this method also to approximate high order derivatives. This fact is very useful if we want to solve *ordinary differential equations* with high order derivatives.
However, only the indirect method provides accurate solutions, as we can see from the following example.
Consider the following differential equation:

$$x^4 y^{(4)} - 4x^3 y^{(3)} + x^2(12 - x^2)y^{(2)} + 2x(x^2 - 12)y^{(1)} + 2(12 - x^2)y = 2x^5 \tag{4.1}$$

in the interval $1 \leq x \leq 11$, subject to the boundary conditions:

$$\begin{cases} y(1) = 1 + e + \frac{1}{e} \\ y(11) = 11 + 121 + 1331 + 11e^{11} + 11e^{-11} \\ y'(1) = 2e \\ y'(11) = 1 + 22 - 363 + 12e^{11} - 10e^{11} \end{cases}$$

The exact solution of (4.1) is:

$$y(x) = x + x^2 - x^3 + xe^x + xe^{-x}$$

We applied both the direct and the indirect method, using Multiquadrics as kernel, choosing $6, 11, 17, 21, 50, 100$ uniformly spaced centers in the interval $[1, 11]$ and $\beta = 7$. After the computation of weights, we test the network on 1001 uniformly spaced target points. We denote with G the matrix for the computation of weights. The matrix is obtained substituting the network approximation of the function and of the derivatives in the differential equation (4.1) and in the boundary conditions. For example in the IRBFN case we have:

$$\begin{aligned} &x^4 \sum_{i=1}^{m} w_i \Phi(x) - 4x^3 \sum_{i=1}^{m+1} w_i H_1(x) + x^2(12 - x^2) \sum_{i=1}^{m+2} w_i H_2(x) \\ &+2x(x^2 - 12) \sum_{i=1}^{m+3} w_i H_3(x) + 2(12 - x^2) \sum_{i=1}^{m+4} w_i H_4(x) = 2x^5 \end{aligned} \tag{4.2}$$

$$\begin{array}{ll} \sum_{i=1}^{m+4} w_i H_4(1) & = 1 + e + \frac{1}{e} \\ \sum_{i=1}^{m+3} w_i H_3(1) & = 2e \\ \sum_{i=1}^{m+4} w_i H_4(11) & = 11 + 11^2 + 11^3 + 11e^{11} + 11e^{-11} \\ \sum_{i=1}^{m+3} w_i H_3(11) & = 1 + 22 + 363 + 12e^{11} - 10e^{-11} \end{array} \tag{4.3}$$

where we denote with H_1, H_2, H_3, H_4 the first, the second, the third and the fourth successive integrations of the kernel Φ.

We denote the first block of matrix G with G_1

$$G_1 = x^4\Phi - 4x^3\mathsf{H}_1 + x^2(12 - x^2)\mathsf{H}_2 + 2x(x^2 - 12)\mathsf{H}_3 + 2(12 - x^2)\mathsf{H}_4 \tag{4.4}$$

where $\Phi, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3, \mathsf{H}_4$ are the matrices obtained from the evaluation of the functions Φ,H_1, H_2, H_3, H_4 in the distance matrix.

Thus we have:

$$G = \begin{pmatrix} G1 \\ cond_1 \\ cond_2 \\ cond_3 \\ cond_4 \end{pmatrix} \tag{4.5}$$

where $cond_1, cond_2$ are the vectors describing the boundary conditions for the function f and $cond_3, cond_4$ are the boundary conditions on the first derivative.

The error is a weighted square-error:

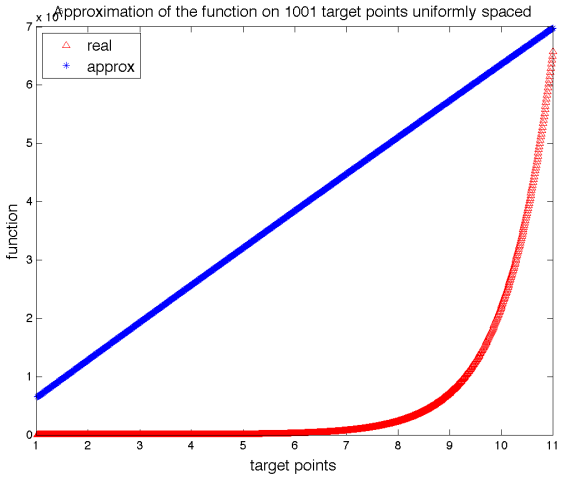
$$ERR = \sqrt{\frac{\sum_{i=1}^{N}(y_i - P_f(x_i))^2}{\sum_{i=1}^{N} y_i^2}} \tag{4.6}$$

where y is the exact solution and P_f its approximation.

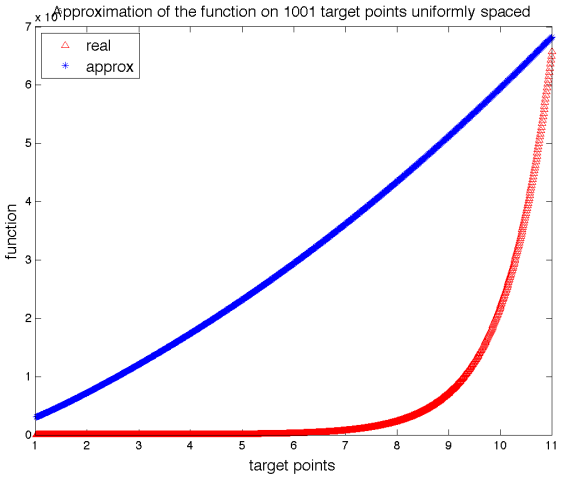
We show the result in Figure 4.1 and we give also a table with the errors and the condition number of the matrix G for different choices of centers. The matlab scripts *DRBFNode.m*, *DRBFNodeTest.m*, *IRBFNode.m*, *IRBFNodeTest.m* contains the implementations of the direct and indirect method for the solution of ODE respectively. The last two contain also the implementation of the Riley method for the study of the condition number (see Section 4.1)

Figure 4.1: Solution of ODE through DRBFN

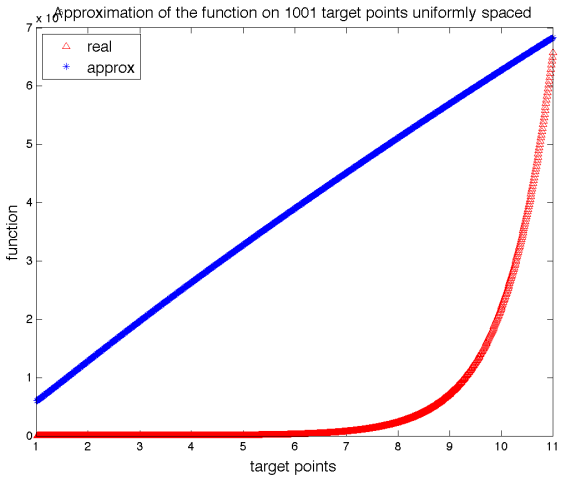
(a) 6 centers



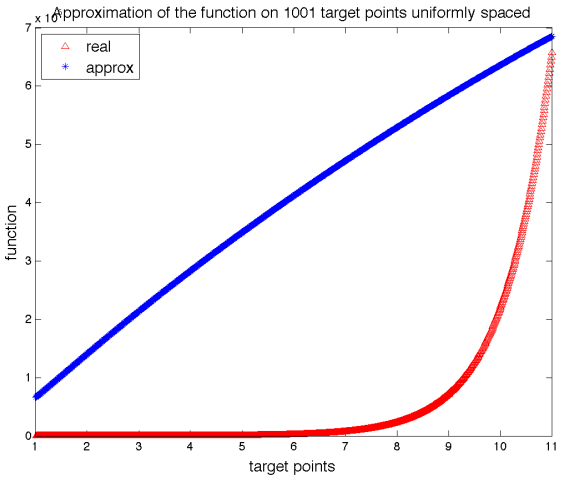
(b) 11 centers



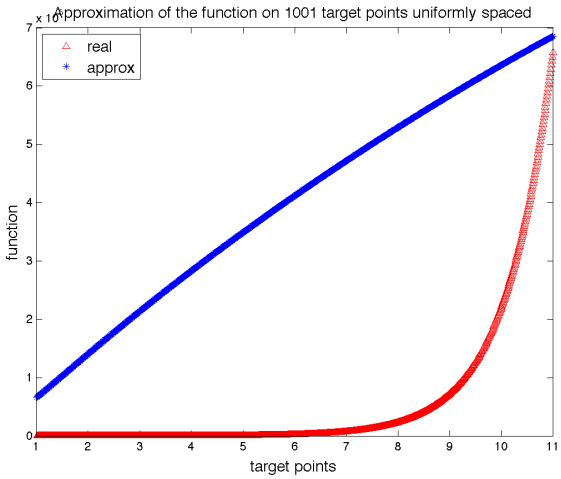
(c) 17 centers



(d) 21 centers



(e) 50 centers



(f) 100 centers

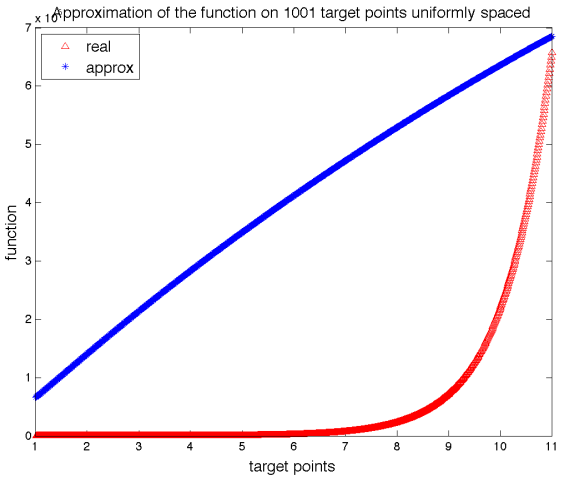


Table 4.1: Error of DRBFN method

Number of centers	Error	Condition Number
6	$2.5e+00$	$1.5e+06$
11	$2.0e+00$	$3.2e+07$
17	$2.5e+00$	$6.4e+07$
21	$2.6e+00$	$7.9e+07$
50	$2.6e+00$	$1.6e+08$
100	$2.5e+00$	$3.7e+08$

As we can see from Figure 4.1 and from the error table, the solution given by the direct method is totally inaccurate. However the condition number of the matrix G is not too large.

We apply now the indirect method; plots are shown in Figure 4.2 and errors in Table 4.2.

Table 4.2: Error of IRBFN method

Number of centers	Error	Condition Number
6	$2.5e-02$	$1.8e+14$
11	$2.2e-04$	$5.5e+12$
17	$6.2e-06$	$1.7e+13$
21	$1.2e-06$	$3.6e+13$
50	$1.4e-07$	$7.7e+14$
100	$7.4e-08$	$9.3e+15$

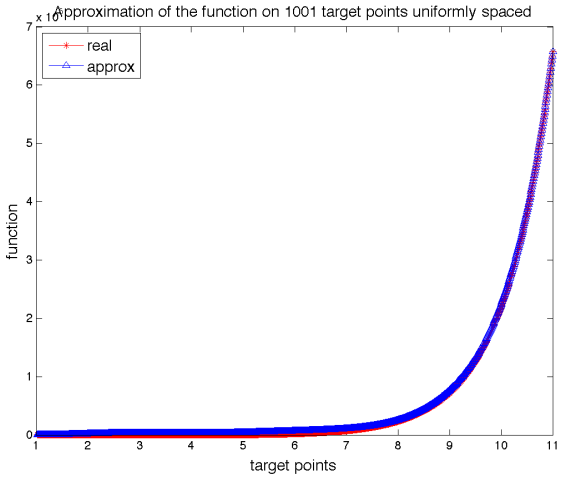
4.1 Analysis of the condition number

From now on, we focus on the analysis of the condition number of the matrix G. As we can see from table 4.1, the direct method is very inaccurate, but the matrix for the computation of weights has not a bad condition number as in the indirect case.

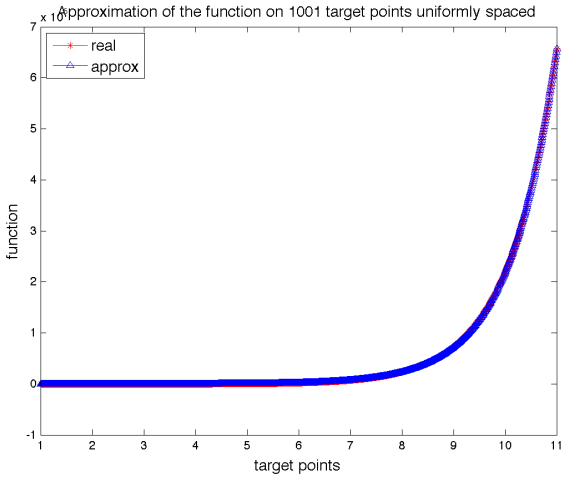
How to lower the condition number? We show two possible solutions.

Figure 4.2: Solution of ODE through IRBFN

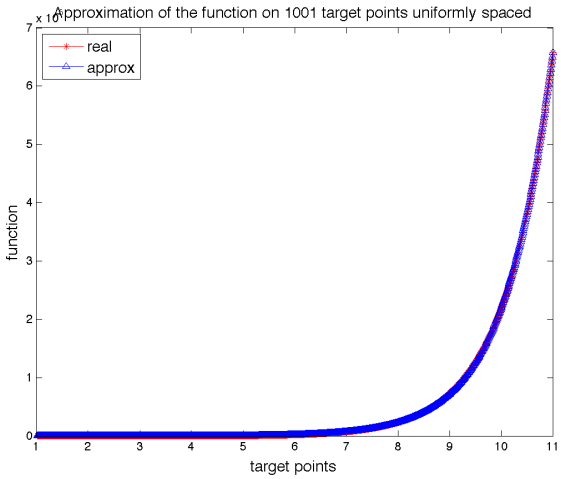
(a) 6 centers



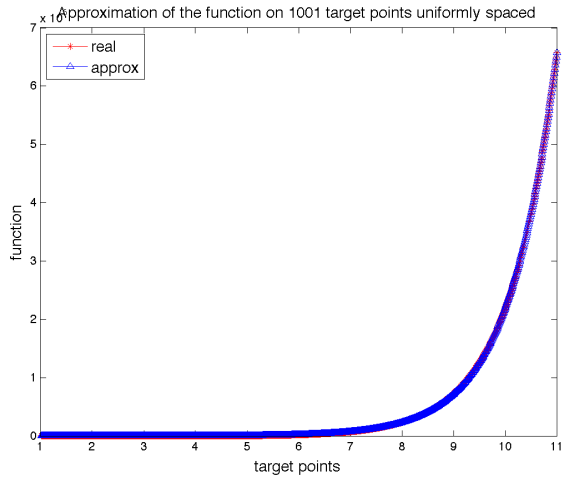
(b) 11 centers



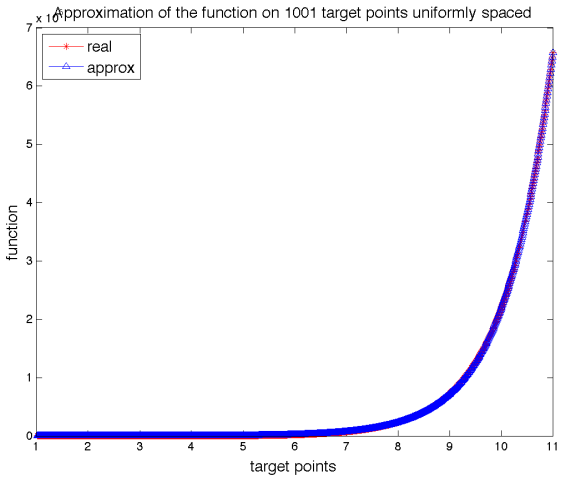
(c) 17 centers



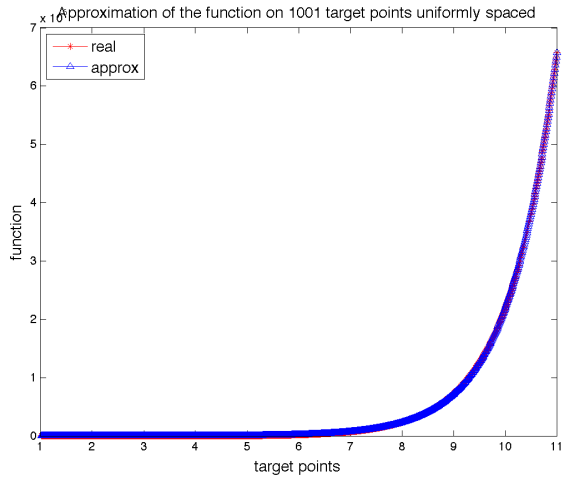
(d) 21 centers



(e) 50 centers



(f) 100 centers



4.1.1 Scaling the domain

The first one consists in *scaling* the domain in which the ordinary differential equation (4.1) is defined, from $1 \leq x \leq 11$ to $\frac{1}{11} \leq x \leq 1$. We also changed the parameter β, and we put $\beta = 0.5$ since we have noticed that this gives a better result.
The implementations of the scaled direct method are contained in the Matlab scripts *DstabOde.m*, *DstabODEtest.m*, while the ones for the indirect method are in the scripts *stabODE.m* and *stabODEtest.m*. In tables 4.3 and 4.4 we display the new error values and new condition numbers.

Table 4.3: Error of DRBFN method

Number of centers	Error	Condition Number
6	$1.8e+00$	$1.3e+03$
11	$8.2e+00$	$1.7e+04$
17	$1.2e+00$	$2.8e+04$
21	$8.6e-01$	$2.1e+04$
50	$1.0e+00$	$1.6e+04$
100	$1.0e+00$	$1.6e+04$

Table 4.4: Error of IRBFN method

Number of centers	Error	Condition Number
6	$6.9e-02$	$4.1e+07$
11	$1.4e-03$	$1.0e+08$
17	$6.2e-04$	$1.6e+08$
21	$3.4e-04$	$2.1e+08$
50	$5.5e-04$	$1.0e+09$
100	$6.9e-04$	$4.3e+09$

Using the scaled method we obtained a better condition number for the matrix G in both the cases. However, this improvement causes a loss of accuracy in the solution. This means that the solution given by the direct method is totally useless, while for the indirect method there is only a small worsening.
We also noticed that the scaling technique works well in this case at least for two reasons: the first is that the definition interval of the ODE is very big ($[1, 11]$) and so the scaling makes a big difference; the second is that the non constant coefficient of the equation are of high order. This causes the matrix G to have entrances like 11^4 and this behaviour leads to a big condition number.
There is no assurance that this technique can work well in completely different cases.

4.1.2 Riley method and iterative methods

The second approach we would try to use to solve the condition problem is by the *Riley method* for linear systems (see [16]).
We illustrate how this method works. Suppose one has to solve a linear system, $Ax = b$ with matrix A which is ill conditioned.
Instead of solving this system we consider the perturbed matrix $C = A + \mu I$ and we solve $Cy = b$. Since $A = C - \mu I$ we can derive $A^{-1} = \frac{1}{\mu}\sum_{k=1}^{\infty}(\mu C^{-1})^k$ and so we can get the solution of the original system as:

$$\begin{aligned} x &= A^{-1}b \\ &= \frac{1}{\mu}\sum_{k=1}^{\infty}(\mu C^{-1})^k b \\ &= \frac{1}{\mu}\sum_{k=1}^{\infty}(\mu C^{-1})^{k-1} y \\ &= y + (\mu C^{-1})y + (\mu C^{-1})^2 y + ... \end{aligned} \tag{4.7}$$

Thus we have $x = y + (\mu C^{-1})y + (\mu C^{-1})^2 y + ...$, that we can rewrite as $x_{k+1} = x_k + (\mu C^{-1})^k y$.
The problem now is: how to choose μ?
In his paper of 1955, Riley (see [16]) suggested to choose $\mu = 10^{(\alpha - p)}$, where p is the desired precision and $\alpha \in \{2, 3\}$.
In any case, to obtain a better condition number for the matrix A, μ should be greater than the minimum eigenvalue of A. On the other hand, for a fast convergence μ should be smaller or equal to the minimum eigenvalue of A.
In our numerical experiments, (see *RileyResiduals.m*, *trial.m*) we try to find the best value for μ using a *trial and error* approach. Moreover, in the algorithm which implements the Riley method we have used the QR factorization instead of the Cholesky one since the matrix is not positive definite.
For a number of centers equal to $6, 11, 17, 21$ the best error value is obtained when μ has the same order of the minimum eigenvalue of A, that means that we have no improvement in the condition number.
Finally we tried to use *iterative methods* applied to the system matrix. Differently from what we did for the Riley method we applied these methods to the scaled problem. As in the case of Riley method we had no improvements in the condition number of the matrix A. This is due mostly to the specific example we are considering. Indeed, in the case we deal with 21 centers, the condition number of the matrix is $2 \cdot 10^8$ (not so high) but the vector of coefficients has a very big norm: $8 \cdot 10^6$. If we compute the relative residual, i.e. $\frac{\|Ax-y\|}{\|y\|}$, in the best case (i.e. using the iterative methods) is $3 \cdot 10^{-11}$ and considering the condition number we can obtain nothing better than what we have with the scaling.

Chapter 5

Application to biological ODEs systems

We want to consider the classical basis function interpolation problem for vector valued functions. This naturally leads to the use of matrix-valued kernels.
The application is the solution of systems of ordinary differential equations.

5.1 Reproducing Kernel Hilbert Spaces of vector-valued functions

We start by stating some elementary results useful to extend the radial basis function theory to the vector-valued case (see [18, 19] for further explanation).
Let $\mathcal{Y}$ be a real Hilbert space with inner product $(\cdot,\cdot)$, $\mathcal{X}$ a set and $\mathcal{H}$ a linear space of functions on $\mathcal{X}$ with values in $\mathcal{Y}$. We assume that $\mathcal{H}$ is also a Hilbert space with inner product $\langle\cdot,\cdot\rangle$.

Definition 5.1 *We say that $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) when for any $y \in \mathcal{Y}$ and $x \in \mathcal{X}$ the linear functional which maps $f \in \mathcal{H}$ to $(y, f(x))$ is continuous.*

In this case according to Riesz lemma, there is for every $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ a function $K(x|y) \in \mathcal{H}$ such that, for all $f \in \mathcal{H}$

$$(y, f(x)) = \langle K(x|y), f\rangle\,. \tag{5.1}$$

Since $K(x|y)$ is linear in y, we write $K(x|y) = K_x y$ where $K_x : \mathcal{Y} \to \mathcal{H}$ is a linear operator. We can rewrite (5.1) as

$$(y, f(x)) = \langle K_x y, f\rangle\,. \tag{5.2}$$

For every $x, t \in \mathcal{X}$ we also introduce the linear operator $K(x,t) : \mathcal{Y} \to \mathcal{Y}$ defined, for every $y \in \mathcal{Y}$, by

$$K(x,t)y = K_t(y)(x) \tag{5.3}$$

We say that $\mathcal{H}$ is *normal* provided there does not exist $(x, y) \in \mathcal{X} \times (\mathcal{Y} \setminus \{0\})$ such that the linear functional $(y, f(x)) = 0$ for all $f \in \mathcal{H}$.
In the following proposition we state the main properties of the function K. To this end, we let $\mathcal{L}(\mathcal{Y})$ be the set of all bounded linear operators from $\mathcal{Y}$ into itself and, for every $A \in \mathcal{L}(\mathcal{Y})$, we denote by A^* its adjoint. We also use $\mathcal{L}_+(\mathcal{Y})$ to denote the cone of

nonnegative bounded linear operators, i.e. $A \in \mathcal{L}_+(\mathcal{Y})$ provided that, for every $y \in \mathcal{Y}$, $(y, Ay) \geq 0$. When this inequality is strict for all $y \neq 0$ we say A is positive definite. Finally, we denote by $\mathbb{N}_m$ the set of positive integers up to and including m.

Proposition 5.1 *If $K(x,t)$ is defined, for every $x, t \in \mathcal{X}$, by equation (5.3) and K_x is given by equation (5.2) the kernel K satisfies for every $x, t \in \mathcal{X}$, the following properties:*

1. *For every $y, z \in \mathcal{Y}$, we have that*

$$(y, K(x,t)z) = \langle K_t z, K_x y \rangle . \tag{5.4}$$

2. *$K(x,t) \in \mathcal{L}(\mathcal{Y})$, $K(x,t) = K(t,x)^*$, and $K(x,x) \in \mathcal{L}_+(\mathcal{Y})$. Moreover, $K(x,x)$ is positive definite for all $x \in \mathcal{X}$ if and only if $\mathcal{H}$ is normal.*

3. *For any $m \in \mathbb{N}$, $\{x_j : j \in \mathbb{N}_m\} \subseteq \mathcal{X}, \{y_j : j \in \mathbb{N}_m\} \subseteq \mathcal{Y}$ we have that*

$$\sum_{j,l \in \mathbb{N}_m} (y_j, K(x_j, x_l) y_l) \geq 0. \tag{5.5}$$

4. *$\|K_x\| = \|K(x,x)\|^{\frac{1}{2}}$.*

5. *$\|K(x,t) \leq \|K(x,x)\|^{\frac{1}{2}} \|K(t,t)\|^{\frac{1}{2}}$.*

6. *For every $f \in \mathcal{H}$ and $x \in \mathcal{X}$ we have that*

$$\|f(x)\| \leq \|f\| \|K(x,x)\|^{\frac{1}{2}}. \tag{5.6}$$

We say that $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is a *kernel* if it satisfies properties 1-3.
So far we have seen that if $\mathcal{H}$ is a RKHS of vector-valued functions, there exist a kernel. In what follows, we show that a kernel determines a RKHS of vector-valued functions.

Theorem 5.2 *If $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is a kernel then there exists a unique (up to an isometry) RKHS which admits K as the reproducing kernel.*

We have to observe that in the case $\mathcal{Y} = \mathbb{R}^n$ the kernel K is a $n \times n$ matrix of scalar-valued functions. The elements of this matrix can be identified by using equation (5.4). Indeed by choosing $y = e_k$ and $z = e_l$, $k, l \in \mathbb{N}_m$ we obtain the following formula

$$(K(x,t))_{kl} = \langle K_x e_k, K_t e_l \rangle . \tag{5.7}$$

In particular when $n = 1$ equation (5.2) becomes $f(x) = \langle K_x, f \rangle$, which is the standard reproducing kernel property.

5.1.1 Interpolation problem for vector-valued functions

Let $f_1, ...f_n : \mathbb{R}^s \to \mathbb{R}$ be mappings which are observed at sampling points $X_i = \{x_{ir} : 1 \leq r \leq N_i\}$ having data sites $D_i = \{d_{ir} : 1 \leq r \leq N_i\}$ such that $f_i(x_{ir}) = d_{ir}$, with $1 \leq r \leq N_i$ and $1 \leq i \leq n$.
What is more we consider the possibility for the functions to be influenced by each other. Thus the generalization of the interpolation problem, also incorporating polynomials is:

$$f_i(x) = \sum_{j=1}^{n} \sum_{r=1}^{N_j} u_{jr} \phi_{i,j}(x, x_{jr}) + \sum_{l=1}^{Q} c_{il} p_l(x) \quad 1 \leq i \leq n \tag{5.8}$$

subject to the side conditions

$$\sum_{r=1}^{N_j} u_{jr} q(x_{jr}) = 0 \quad \forall q \in \pi_{k-1}^s,\ 1 \leq j \leq n \tag{5.9}$$

where π_{k-1}^s is the space of polynomials in $\mathbb{R}^s$ of total degree not exceeding $k-1$. Within this model the kernel $\phi_{i,j}$ expresses the influence of the jth component f_j on the ith.
If we interpret the functions f_i, $1 \leq i \leq n$, not separately, but as components of one single vector-valued function $f : \mathbb{R}^s \to \mathbb{R}^n$ we can state the interpolation problem for vector valued data.

Problem 5.3 *Let $n \in N$ and $N_i \in \mathbb{N}$ for $1 \leq i \leq n$. Given n sets $X_i = \{x_{ir} : 1 \leq r \leq N_i\} \subset \mathbb{R}^s$ of data points and n vectors of values observed at these points $d_i \in \mathbb{R}^{N_i}$, solving the interpolation problem*

$$f_i(x_{ir}) = d_{ir} \quad 1 \leq r \leq N_i, \quad 1 \leq i \leq n$$

we obtain the linear system:

$$\begin{pmatrix} \psi_{1,1} & \psi_{1,2} & \cdots & \psi_{1,n} & P_1 & O & \cdots & O \\ \psi_{2,1} & \psi_{2,2} & \cdots & \psi_{2,n} & O & P_2 & \cdots & O \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \psi_{n,1} & \psi_{n,2} & \cdots & \psi_{n,n} & O & O & \cdots & P_n \\ P_1^T & O & \cdots & O & O & O & \cdots & O \\ O & P_2^T & \cdots & O & O & O & \cdots & O \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & P_n^T & O & O & \cdots & O \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{5.10}$$

where

$$\begin{aligned}
&\psi_{i,j} = [\phi_{i,j}(x_{i,r}, x_{j,s})]_{r,s=1}^{N_i,N_j} \in \mathbb{R}^{N_i \times N_j} \quad 1 \leq i,j \leq n \\
&P_i = [p_l(x_{ir})]_{r,l=1}^{N_i,Q} \in \mathbb{R}^{N_i \times Q} \qquad 1 \leq i \leq n \\
&u_i = [u_{ir}]_{r=1}^{N_i} \in \mathbb{R}^{N_i} \qquad 1 \leq i \leq n \\
&c_i = [c_{il}]_{l=1}^{Q} \in \mathbb{R}^{Q} \qquad 1 \leq i \leq n \\
&d_i = [d_{ir}]_{r=1}^{N_i} \in \mathbb{R}^{N_i} \qquad 1 \leq i \leq n
\end{aligned} \tag{5.11}$$

We can show that under some conditions on the matrix of kernels the interpolation problem has a unique solution. Therefore, we have to transfer the notion of conditionally positive definiteness into the setting of matrices of kernels.
We define the following subspace:

$$V_{X,k} = \left\{ [p(x_r)]_{r=1}^{N} : p \in \pi_{k-1}^s \right\} \subset \mathbb{R}^N \tag{5.12}$$

where $X = \{x_r : 1 \leq r \leq N\} \subset \mathbb{R}^s$ is a unisolvent set of points.
Thus, the dimension of $V_{X,k}$ is equal to the dimension of π_{k-1}^s.
We can now give the definition of *matrix conditionally positive definite kernels.*

Definition 5.2 *An $n \times n$ matrix of kernels $\Psi = [\phi_{i,j}]_{i,j=1}^n$ with $\phi_{i,j} : \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{R}$ is said to be matrix conditionally positive definite of order k on $\mathbb{R}^s$ if for any n sets of distinct points $X_i = \{x_{ir} : 1 \leq r \leq N_i\} \subset \mathbb{R}^s$ the matrix $\mathbf{\Psi} = [\psi_{i,j}]_{i,j=1}^n$, where*

$$\psi_{i,j} = [\phi_{i,j}(x_{i,r}, x_{j,s})]_{r,s=1}^{N_i,N_j} \in \mathbb{R}^{N_i \times N_j} \quad 1 \leq i,j \leq n$$

is conditionally positive semidefinite with respect to the subspace $V_{X_1,k} \times \cdots \times V_{X_n,k}$. The matrix of kernels is said to be strictly matrix conditionally positive definite of order k on $\mathbb{R}^s$ if for any n sets of distinct points $X_1, ..., X_n$ the matrix $\mathbf{\Psi}$ is conditionally positive definite with respect to $V_{X_1,k} \times \cdots \times V_{X_n,k}$.

where Ψ is the vector-valued kernel, $\phi_{i,j}$ is the scalar-valued kernel, $\mathbf{\Psi}$ is the evaluation of Ψ in a proper matrix of distances.
The following proposition shows that strictly matrix conditionally positive definiteness is sufficient to guarantee existence and uniqueness of the solution of the corresponding interpolation problems.

Proposition 5.1 *Let $n \in \mathbb{N}$ and suppose that the $n \times n$ matrix of kernels $\Psi = [\phi_{i,j}]_{i,j=1}^n$ is strictly conditionally positive definite of order k on $\mathbb{R}^s$. Suppose that the n sets $X_i = \{x_{ir} : 1 \leq r \leq N_i\}$ of distinct points of $\mathbb{R}^s$ are all unisolvent for π_{k-1}^s. Then the matrix of (5.10) is invertible and the interpolation problem has a unique solution for any given set of values $d_i \in \mathbb{R}^{N_i}$, $1 \leq i \leq n$.*

Proof
Consider the homogeneous system corresponding to Problem (5.3). Let $\mathbf{u}^T = [u_1^T, ..., u_n^T]$, $\mathbf{c}^T = [c_1^T, ..., c_n^T]$ and $[\mathbf{u}^T, \mathbf{c}^T]$ be a solution of the homogeneous system. Then, from the lower part of the block system (5.10) we have that

$$P_i^T u_i = 0 \quad 1 \leq i \leq n \tag{5.13}$$

which shows that u_i belongs to $V_{X,k}^{\perp}$, for $1 \leq i \leq n$. From the first rows of the block system (5.10) we obtain that

$$\begin{pmatrix} \psi_{1,1} & \cdots & \psi_{1,n} \\ \vdots & & \vdots \\ \psi_{n,1} & \cdots & \psi_{n,n} \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} + \begin{pmatrix} P_1 c_1 \\ \vdots \\ P_n c_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \tag{5.14}$$

Multiplying on the left by $\mathbf{u}^T$ leads to

$$\sum_{i=1}^{n} \sum_{j=1}^{n} u_i^T \psi_{i,j} u_j + \sum_{j=1}^{n} (P_j^T u_j)^T c_j = 0. \tag{5.15}$$

Thus, condition (5.13) implies that $\mathbf{u}^T \boldsymbol{\Psi} \mathbf{u} = 0$. However, by hypothesis the matrix

$$\boldsymbol{\Psi} = (\psi_{i,j})_{i,j=1}^{n} = ([\phi_{i,j}(x_{i,r}, x_{j,s})]_{r,s=1}^{N})_{i,j=1}^{n} \tag{5.16}$$

is positive definite on the subspace $V_{X_1,k}^{\perp} \times \cdots \times V_{X_n,k}^{\perp}$.
Hence the quadratic form can only be zero if $\mathbf{u} = 0$. Thus the block system (5.14) reduces to

$$P_j c_j = 0, \quad j = 1, ..., n. \tag{5.17}$$

Now since the sets X_i are unisolvent, we conclude that the vectors c_j, $1 \leq i \leq n$ are zero. Therefore, the only solution of the homogeneous system is the trivial solution, and the matrix of Problem (5.3) is invertible.

□

We have shown that existence and uniqueness of the solution to the interpolation Problem 5.3 is guaranteed for strictly matrix conditionally positive definite kernels.
We now derive **sufficient** conditions on the matrix component kernels $\phi_{i,j} : \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{R}$ which provide strictly matrix conditionally positive definite kernels.
For this purpose we introduce the notation $\tilde{\phi}(x, y) = \phi(y, x)$.

Theorem 5.4 *Let $n \in \mathbb{N}$ and $\phi_{i,j} : \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{R}$, $1 \leq i, j \leq n$, be kernels. Suppose that:*

1. *the matrix $S = [\sigma_{i,j}]_{i,j=1}^{n}$ is symmetric with entries in $\{\pm 1\}$,*

2. *the kernels $\sigma_{i,j}(\phi_{i,j} + \tilde{\phi}_{i,j})$ are symmetric and conditionally positive definite of order k for all $1 \leq i \neq j \leq n$, and*

3. *the kernels* $(2\phi_{i,i} - \sum_{j\neq i} \sigma_{i,j}(\phi_{i,j} + \tilde{\phi}_{i,j}))$ *are strictly conditionally positive definite of order* k *for all* $1 \leq i \leq n$.

Then the $n \times n$ *matrix of kernels* $\Phi = [\phi_{i,j}]_{i,j=1}^{n}$ *is strictly conditionally positive definite of order* k *on* $\mathbb{R}^s$.

The theorem proof makes use of two lemma.

Lemma 5.1 *For any matrix* $G \in \mathbb{R}^{n\times n}$ *conditionally (positive or negative) semidefinite with respect to subspace* $V \subset \mathbb{R}^n$ *we have that*

$$|u^T G v + v^T G u| \leq |u^T G u + v^T G v|, \quad for\ all\ u, v \in V^{\perp}. \tag{5.18}$$

With this lemma we can show positive definiteness of a block matrix with the same structure as the upper left block of the matrix of system (5.10).

Lemma 5.2 *Let* $A_{ij} \in \mathbb{R}^{N\times N}$ *for* $1 \leq i, j \leq n$ *and let* V *be a subspace of* $\mathbb{R}^N$. *Suppose that*

1. *the matrix* $S = [\sigma_{i,j}]_{i,j=1}^{n}$ *is symmetric with entries in* $\{\pm 1\}$,
2. *the matrices* $\sigma_{i,j}(A_{ij} + A_{ij}^T)$ *are symmetric and conditionally positive semidefinite with respect to* V *for* $j \neq i$, *and*
3. *the matrices* $(2A_{i,i} - \sum_{j\neq i} \sigma_{i,j}(A_{ij} + A_{ij}^T))$ *are conditionally positive definite with respect to* V *for* $1 \leq i \leq n$.

Then the matrix

$$A := [A_{ij}]_{i,j=1}^{n} \in \mathbb{R}^{nN\times nN}$$

is conditionally positive definite with respect to the subspace $V^n \subset \mathbb{R}^{nN}$.

We observe that, for the case $N = 1$ in Lemma 5.2 with $V = \{0\}$ 2 is automatically fulfilled and 3 is a statement about the dominance of diagonal entry A_{ii}. Thus, the lemma can be interpreted as a Gershgorin type statement for block matrices.

In what follows we are interested in matrix kernels defined by the product of a single scalar-valued kernel with a matrix of mixture coefficients. We state a theorem, that in this particular case gives a sufficient condition for matrix conditional positive definiteness.

Theorem 5.5 *Let* $C \in \mathbb{R}^{n\times n}$ *be a positive semidefinite matrix and* $\phi : \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{R}$ *a conditionally positive definite kernel of order* k. *Furthermore, let at least one of* C *or* ϕ *be symmetric. Then*

$$\Psi = C\Psi = [c_{ij}\psi]_{i,j=1}^{n} : \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{R}^{n\times n}$$

is a matrix conditionally positive definite kernel of order k. *If* C *is positive definite and* $\phi : \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{R}$ *is strictly conditionally positive definite of order* k, *then* Ψ *is strictly matrix conditionally positive definite of order* k.

We remark that we do not expect the conclusion of Theorem 5.5 to hold if neither Ψ nor C is symmetric. However, sometimes the use of a non-symmetric matrix C gives better results than in the symmetric case (see [19]).

5.2 Solution of ODEs systems

In the following we show the results obtained by applying the indirect approach (IRBFN) to a system of ODEs. The solution technique is based on the vector-valued kernel theory, developed in the previous section.
In all our numerical experiments we consider a matrix C such that:

$$C = \begin{pmatrix} 1 & \alpha \\ \beta & 1 \end{pmatrix}. \tag{5.19}$$

According to Theorem (5.5) the matrix C should be positive definite to guarantee matrix kernel strictly conditionally positive definiteness. If the matrix is symmetric, by applying Schur complement condition for positive definiteness we obtain that $\alpha\beta < 1$. If we require only conditionally positive definiteness, for our matrix, then we can consider a non strict inequality $\alpha\beta \leq 1$.
However, it is also true that if a matrix is symmetric and diagonally dominant then is positive semidefinite. Moreover if a matrix is diagonally dominant there is no need for (partial) pivoting when performing Gaussian elimination and we are sure that the Jacobi and Gauss Seidel methods converge. Thus, we ask $|\alpha + \beta| < 2$.

The first example is a system of two coupled differential equations of the first order:

$$\begin{cases} x' = & 3x + y \\ y' = & -x + 5y \end{cases} \tag{5.20}$$

With initial conditions $x(0) = 1$ and $y(0) = 0$.
Clearly, it is possible to analytically determine the solution of (5.20) that is:

$$\begin{array}{ll} x = & (1-t)e^{4t} \\ y = & (-t)e^{4t} \end{array} \tag{5.21}$$

In order to solve the system, we have to substitute the expression, which defines the network, in both the equations of (5.20), as we did for the differential equation (4.2). Then, we have to write the system in a matrix form like (5.10). Finally we multiply this system for the matrix C.
The solution of the system (5.20) and the study of the parameter for the matrix C is implemented in the Matlab script *OdeSystem.m*. Since it is possible to analytically compute the solution we can calculate the l_2-error comparing the approximation with the exact solution.

We choose randomly $\alpha, \beta \in [-0.9, 0.9]$. For symmetric case, we mean that $\alpha = \beta$.
In Table 5.1 we display the minimum of the error for different choices of the parameters, while in Figure 5.1 we display the error plots.

Table 5.1: Error solution ODEs system

	x-component	y-component
Non-symmetric	$9.6e+00$	$9.5e+01$
Symmetric	$1.2e+01$	$9.5e+01$

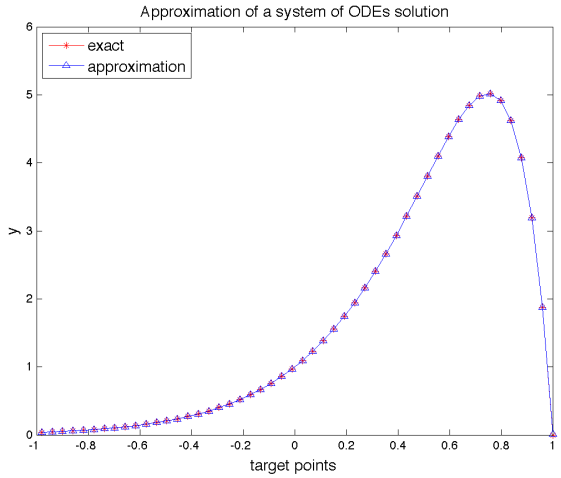
In Figure 5.2 we show the approximation of the function when $\alpha = \beta = 1$. In this case the l_2-error is displayed in Table 5.2:

Table 5.2: Error solution ODEs system

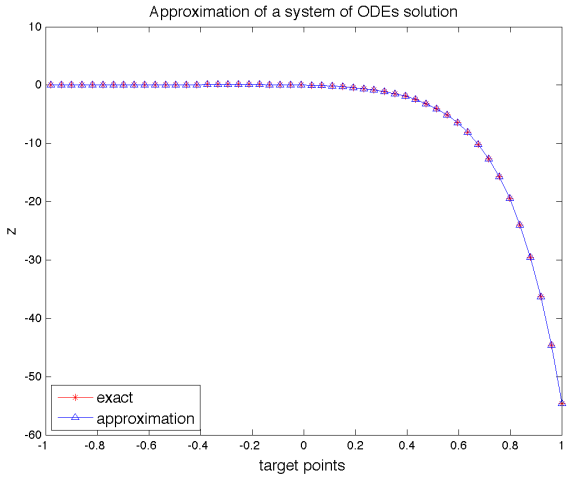
x-component	y-component
$9.3e-03$	$2.6e-02$

Figure 5.2: Approximation of the solution with $\alpha = \beta = 1$

(a) Approximation of the x-component



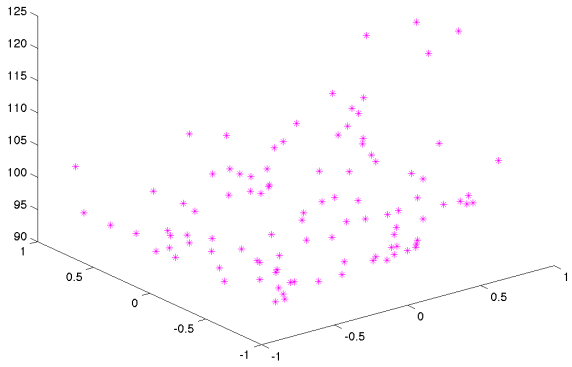
(b) Approximation of the y-component



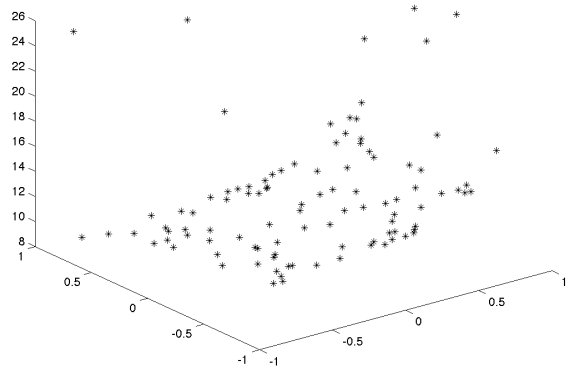
If $\alpha = \beta = 1$ the condition on matrix C to be diagonally dominant is not satisfied. However the matrix C is semidefinite positive and thus from theorem 5.5 we are guaranteed to have a positive definite kernel matrix.
Moreover, this choice of the parameter completely reflects how the equations are coupled.

Figure 5.1: 3D plots of the error for different choices of α and β

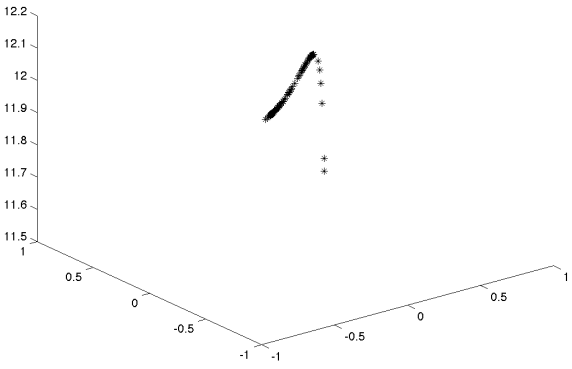
(a) Error of the x-component non-symmetric case



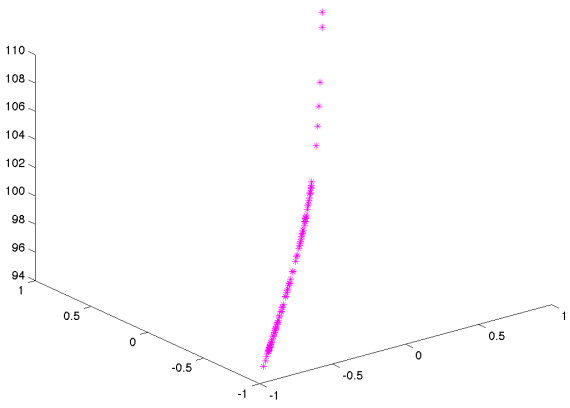
(b) Error of the y-component non-symmetric case



(c) Error of the x-component symmetric case



(d) Error of the y-component symmetric case

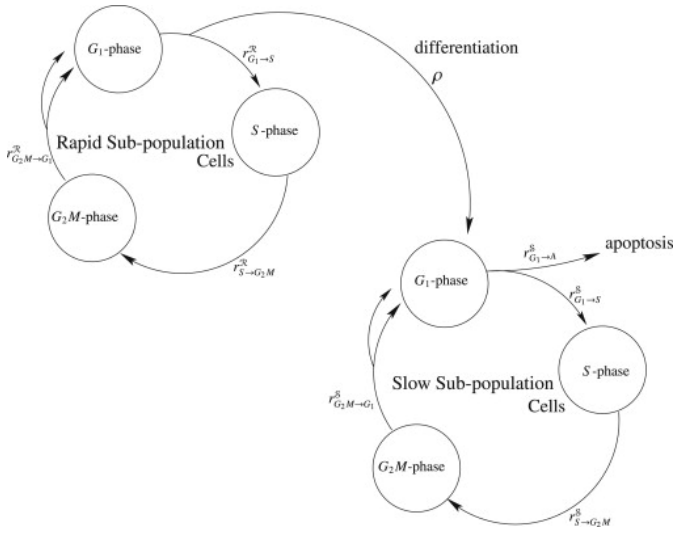


5.2.1 Biological Model

In this section we apply our numerical method for the solution of ODEs, to a biological model for diabetes and insulin therapy.
Diabetes mellitus is a disease of the glucose regulatory system characterized by fasting and/or postprandial hyperglycemia. Different factors are involved in the developing of this disease and the model we are taking into account is the union of three different partial differential models; the first one describes the insulin storage and secretion (see [20]); the second one the glucose and insuline dynamics and finally the third one models β-cell cycle (see [21, 22]).
In the following we solve the one concerning β-cell cycle. This is a *multi-compartment model*, which is a mathematical model used for describing the way materials or energies are transmitted among the compartments of a system. In our specific case the compartments are represented by the cell cycle phases and the transmission is represented by the different transition rates.

β-cells are responsible for the production and the store of insulin. They are directly connected with the blood circuit and so they are able to measure the glucose concentration with great accuracy.
The cell-cycle of β-cells consists of different phases that lead finally to cell division, the mitosis.
In the G_1 phase the cell grows and some cell components are added. The S phase is the synthetis phase in which the DNA reduplicates. In the G_2 phase the cell prepares for mitosis and in the M phase the division of the chromosomes, the nucleus and the cell takes place. The two phases G_2 and M are considered together and they form the phase G_2/M.

Figure 5.3: Cell cycle phases



In the model the transition rates c_1, c_2, c_3 and the apoptosis rate, which we indicate with μ_A, are also considered.
Thus, the cell-cycle can be modelled as a three compartmental model:

$$\begin{aligned} \tfrac{dG_1(t)}{dt} &= 2c_3 \cdot G_2/M(t) - (c_1 + \mu_A) \cdot G_1(t) \\ \tfrac{dS(t)}{dt} &= c_1 \cdot G_1(t) - c_2 \cdot S(t) \\ \tfrac{dG_2/M(t)}{dt} &= c_2 \cdot S(t) - c_3 \cdot G_1(t) \end{aligned} \tag{5.22}$$

where $G_1(t)$, $S(t)$ and $G_2/M(t)$ are the number densities of cells in G_1, S and G_2/M phase respectively.
Insulin plays an important role in regulating β-cell mass. To model the effect of insulin on the cell cycle we introduce a new transition rate

$$c_1^*(t) = c_1 + rI(t) \cdot c_1 \tag{5.23}$$

where c_1 is the original transition rate, r is a constant which describes the magnitude of the insulin effect on the cell cycle and $I(t)$ is the insulin concentration in the blood at time t.
Thus, the model, including the insulin influence on the cell cycle is the following

$$\begin{aligned} \tfrac{dG_1(t)}{dt} &= 2c_3 \cdot G_2/M(t) - ((c_1 + rI(t)c_1) + \mu_A) \cdot G_1(t) \\ \tfrac{dS(t)}{dt} &= (c_1 + rI(t)c_1) \cdot G_1(t) - c_2 \cdot S(t) \\ \tfrac{dG_2/M(t)}{dt} &= c_2 \cdot S(t) - c_3 \cdot G_1(t) \end{aligned}$$

We rewrite (5.2.1) in a more synthetic way, by the following substitution:

$$\begin{cases} G_1(t) = x_1(t) \\ S(t) = x_2(t) \\ G_2/M(t) = x_3(t) \\ I(t) = x_5(t) \\ c_1 = p_1 \\ c_2 = p_2 \\ c_3 = p3 \\ \mu_A = p_4 \\ r = p_5 \end{cases} \tag{5.24}$$

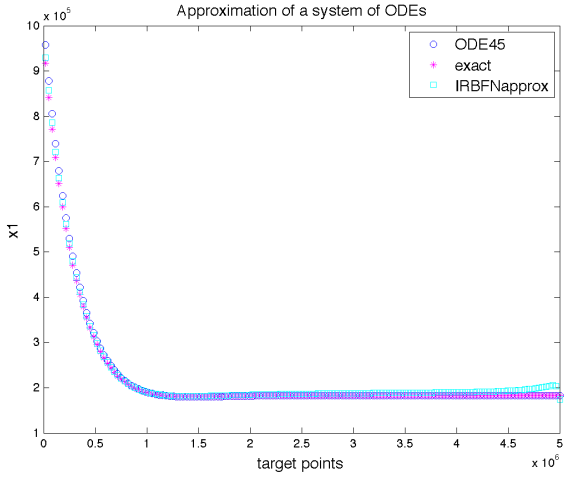
Thus, the system is:

$$\begin{cases} x_1' = & 2p_3x_3 - (p_1(1+p_5\bar{x}_5)+p_4)x_1 \\ x_2' = & p_1(1+p_5\bar{x}_5)x_1 - p_2x_2 \\ x_3' = & p_2x_2 - p_3x_3 \end{cases} \tag{5.25}$$

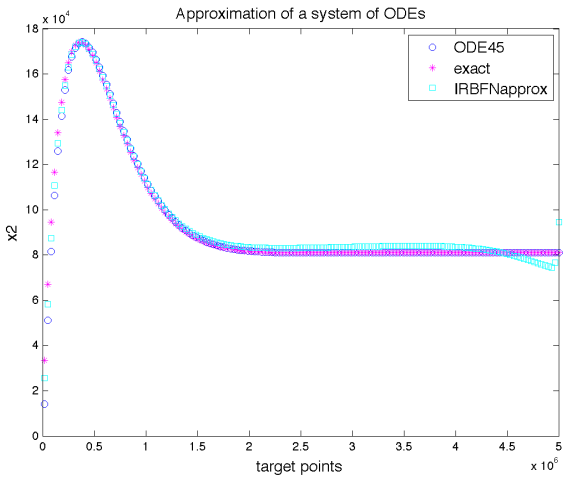
We have to estimate x_1, x_2 and x_3. The parameter p_1, p_2, p_3, p_4, p_5 are determined through model analysis. To obtain a unique solution we should choose a value for the parameter $\bar{x}_5$. We decided to take $\bar{x}_5 = \frac{p_4 - p_1}{p_1 p_5}$. With this choice the model is in its steady state.

Figure 5.4: Approximation of the cell_cycle solution

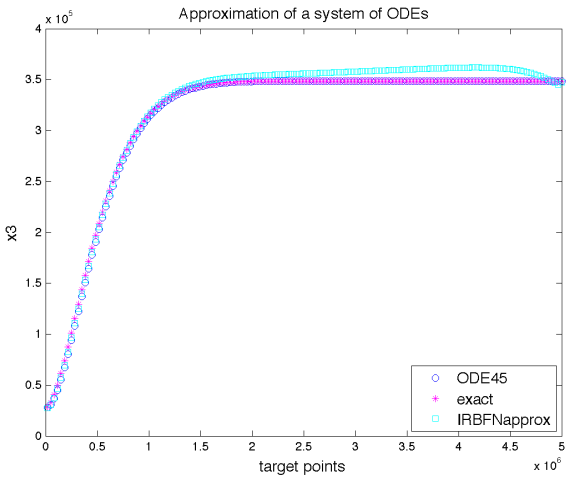
(a) Approximation of the x_1-component



(b) Approximation of the x_2-component



(c) Approximation of the x_3-component



The implementation is in the Matlab script *cell_cycle_res.m* which uses the Matlab function *cell_cycle.m*. In Figure 5.4, we compare the exact solution, the approximation

obtained by using Matlab solver ODE45 and the solution obtained through IRBFN. In Table 5.3 we display the weighted square error (4.6) for the three components by using the IRBFN method.

Table 5.3: Error solution cell-cycle ODEs system

	x_1-component	x_2-component	x_3-component
IRBFN	$2.8e-02$	$2.9e-02$	$2.5e-02$
ODE45	$3.1e-02$	$2.8e-02$	$7.1e-03$

5.3 Conclusions

In this thesis we aimed to study a method based on a Radial Basis Function Network (RBFN) to solve a system of ODEs in order to apply it to a biological model for diabetes.

After introducing the theory of Radial Basis Functions (RBF) and Artificial Neural Networks (ANN), we focused on the application of the RBFN to the simultaneous approximation of a function and its derivatives. In particular, we provided a deep analysis on the functioning of the method by using different kernels. Our original contribution was aimed to understand why some RBF provide better accuracy than others. This fact has been explained by some observations on the localization properties of different kernels.
We also noticed that a difference in accuracy may depend on the function we want to approximate. Indeed, it is fundamental that this function belongs to the native space of the kernel.

The following step was to use the RBFN to solve an ordinary differential equation of arbitrary order. This led to face with the problem of the condition number of the matrix which computes the weights during the training phase of the network.
We found that a scaling of the domain in which the ODE is defined, provides a great reduction in the condition number and that, even if this is a very simple technique it has been proven to work much better with respect to the Riley method.

Finally, we reached our purpose of approximating a system of ODEs by a synergistic use of the vector valued kernel theory and of RBFN. Indeed, we interpreted the problem of the solution of a ODEs system as an interpolation problem of a vector valued function, whose components represent the differential equations which compose the system.
In order to obtain a well posed interpolation problem, we have introduced the concept of a strictly conditionally positive kernel matrix. If we want to obtain such a kernel matrix from a strictly conditionally positive definite scalar valued kernel ϕ, it is sufficient to multiply the matrix, obtained as the evaluation of the kernel ϕ in the matrix of distances, for a strictly positive definite matrix of coefficients, which we indicate with C.

We made a parametric analysis in order to determine the coefficients of the matrix C on the basis that a strictly diagonally dominant matrix is also positive definite. We then applied the RBFN to the solution of a system of two coupled differential equations and to the model for the β-cell cycle.
It is worth noting that, even if the error in the approximation by using RBFN is almost of the same order of the Matlab toolbox ODE45, our method is not a *black box* for the solution of ordinary differential equation. We expect that by improving the parameter analysis and also by using different types of kernels our results would improve.

5.3.1 Future directions

In all our numerical experiments we did not pay enough attention to the choice of the centers and of the data sites, and we have always considered equally spaced centers and target points. However, since a different location of the points may reduce the error in the boundary of the domain or may lead to an increase in the interpolation accuracy, this could be object of further studies.
Another limitation of the work is in the numerical solution of the linear system which appears in the determination of the weights: it may be possible to make the process faster by using different basis as Newton basis.
Moreover, a lot of work has to be done in order to improve the method for solving the system of ODEs.

- by changing the method for the parameter analysis of the matrix C;
- by using different kernels to determine the vector-valued kernel matrix and not only the Multiquadrics;
- by applying the method to the whole system of differential equations.

This investigation will be the main direction the authors will pursue in the near future.

Appendix A

DRBFN and IRBFN in two dimensions

For the sake of completeness, we show that DRBFN and IRBFN can be also applied in the two dimensional case.

Concerning the DRBFN, there are no differences with respect to the one dimensional case. Thus, we concentrate on the IRBFN.

Let us take $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, we approximate f and all its derivatives with respect to variable x_1 and x_2.

As in the one dimensional case, the key idea of the method is that we can simultaneously approximate a function and its derivatives up to order n, simply approximating the nth derivative with (3.9) and then computing n integrations.

However, the way we compute the basis functions is slightly different.

For example, consider $f(x_1, x_2) = x_1^2 x_2 + \frac{x_2^3}{3} + \frac{x_2^2}{2}$ in the square $[-3, 3]^2$. We denote with $\mathbf{x} = [x_1, x_2]^T$ the data sites column vector and with $\mathbf{c} = [c_1, c_2]^T$ the centers column vector. We suppose there are m centers and n data sites.

We approximate the second partial derivative of function f with respect to the variable x_1 as follows:

$$\frac{\partial^2 f(\mathbf{x}_j)}{\partial x_1^2} \approx \sum_{i=1}^{m} w_i \Phi(\mathbf{x}_j - \mathbf{c}_i) \quad \forall\, j = 1, ..., n \tag{A.1}$$

Then, the first partial derivative with respect to variable x_1 is

$$\begin{aligned} \frac{\partial f(\mathbf{x}_j)}{\partial x_1} &= \int \frac{\partial^2 f(\mathbf{x}_j)}{\partial x_1^2} dx_1 \\ &\approx \sum_{i=1}^{m} w_i \int \Phi(\mathbf{x}_j - \mathbf{c}_i) dx_1 \\ &= \sum_{i=1}^{m} w_i H(\mathbf{x}_j) + C_1(x_2) \end{aligned} \tag{A.2}$$

where H is the first integration of the kernel $\Phi(\mathbf{x})$ with respect to the variable x_1.

Finally, the function estimate is

$$\begin{aligned} f(\mathbf{x}_j) &= \int \frac{\partial f(\mathbf{x}_j)}{\partial x_1} dx_1 \\ &\approx \sum_{i=1}^{m} w_i \int H(\mathbf{x}_j) dx_1 \\ &= \sum_{i=1}^{m} w_i \bar{H}(\mathbf{x}_j) + C_1(x_2)x_1 + C_2(x_2) \end{aligned} \tag{A.3}$$

where $\bar{H}$ is the second integration of the kernel $\Phi(\mathbf{x})$ with respect to the variable x_1.

Thus, we have to determine the vector of the weights w_i, but we have also to approximate the functions $C_1(x_2)$ and $C_2(x_2)$. This can be done by using the one dimensional IRBFN, i.e. considering the following formula:

$$C_1(x_2) = \sum_{i=1}^{M} \mathsf{w}_i \mathsf{H}(x_2) + K_1 x_2 + K_2 \tag{A.4}$$

where H is the second integration of the one-dimensional kernel $\Phi(x_2)$ and K_1 and K_2 are the integration constants of the one-dimensional IRBFN (see 3.12).

An analogous formula can be used for the function $C_2(x_2)$.

As in the one-dimensional case, we are interested in using Gaussians, Inverse Multiquadrics and Multiquadrics as kernels.

The implementations are in the Matlab functions *DRBFN2Dtest.m* and *IRBFN2Dtest.m* (which are used by the Matlab GUI *mygui.m*) which test the network by using the Matlab functions *DRBFN2D.m* and *IRBFN2D.m* to compute the weights.

In Table A.1 we display the l_2-error for different kernels and for the approximation of the functions and its first and second partial derivatives with respect to variable x_1 by using DRBFN; in Table A.2 we show the l_2-error for IRBFN.

Table A.1: Error of DRBFN method

	f	$\partial f/\partial x_1$	$\partial^2 f/\partial x_1^2$
G	$1.1e-02$	$1.0e-02$	$1.0e-01$
IM	$6.0e-03$	$1.6e-02$	$2.4e-01$
M	$4.0e-03$	$1.5e-02$	$1.9e-01$

Table A.2: Error of IRBFN method

	f	$\partial f/\partial x_1$	$\partial^2 f/\partial x_1^2$
G	$2.1e-03$	$1.0e-03$	$2.3e-03$
IM	$6.5e-03$	$1.2e-03$	$2.8e-03$
M	$8.3e-04$	$4.9e-04$	$6.1e-04$

In Table A.3 we display the l_2-error for different kernels and for the approximation of the functions and its first and second partial derivatives with respect to variable x_2 by using DRBFN; in Table A.4 we show the l_2-error for IRBFN.

Table A.3: Error of DRBFN method

	f	$\partial f/\partial x_2$	$\partial^2 f/\partial x_2^2$
G	$1.1e-02$	$4.6e-03$	$4.8e-02$
IM	$6.0e-03$	$2.4e-02$	$3.5e-01$
M	$4.0e-03$	$1.1e-02$	$1.3e-01$

Table A.4: Error of IRBFN method

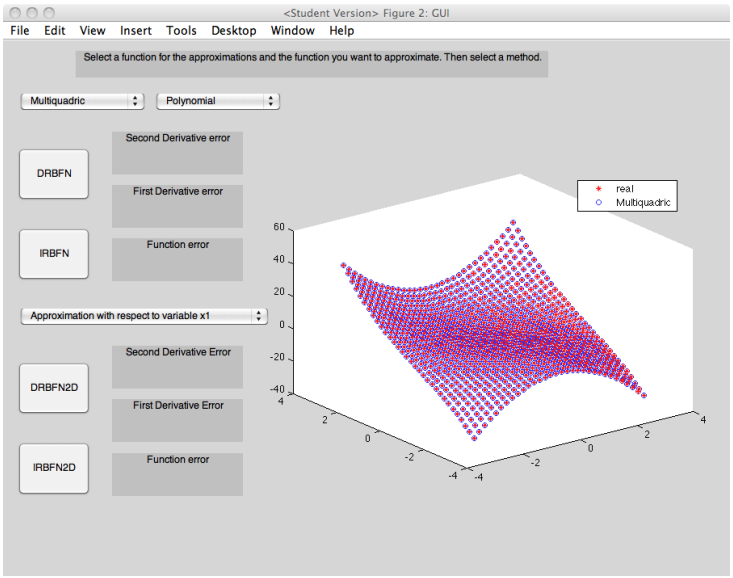
	f	$\partial f/\partial x_2$	$\partial^2 f/\partial x_2^2$
G	$3.5e-04$	$6.4e-04$	$5.6e-03$
IM	$2.3e-03$	$9.0e-04$	$1.1e-02$
M	$1.4e-03$	$7.3e-04$	$1.3e-03$

Appendix B

A GUI interface with Matlab

We have developed a GUI interface with Matlab which displays the plots for various functions approximation with DRBFN and IRBFN even in the two dimensional case.

Figure B.1: A Matlab GUI



The implementations are in the Matlab functions *MayDRBFNtestGUI.m*, *MayIRBFN-testGUI.m*, *MayDRBFN.m*, *MayIRBFN.m*, *DRBFN2Dtest.m*, *IRBFN2Dtest.m*, *DRBFN2D.m*, *IRBFN2D.m* and in the Matlab script *mygui.m*.

Bibliography

[1] Gregory E. Fasshauer, *Meshfree Approximation Methods with Matlab*, Interdisciplinary Mathematical Sciences, Vol.6. (Cited on pages 7, 10, 39 and 46.)

[2] Scott A. Sarra, Edward J. Kansa, *Multiquadric Radial Basis Function Approximation Methods for the Numerical Solution of Partial Differential Equations*. (Cited on pages 1 and 18.)

[3] Dario Floreano, Claudio Mattiussi, *Manuale sulle reti neurali*, Il Mulino, 2002. (Cited on page 27.)

[4] Simon Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall International, 1999. (Cited on page 27.)

[5] D. S. Broomhead, David Lowe, *Multivariable Functional Interpolation and Adaptive Networks*, Complex Systems, Vol. 2, pp.321-355(1988). (Cited on pages 29 and 32.)

[6] Nam Mai-Duy, Thanh Tran-Cong, *Approximation of function and its derivatives using radial basis function networks*, Applied Mathematical Modelling, 27, pp. 197-220(2003) . (Cited on pages 2 and 35.)

[7] Holger Wendland, *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics(2005). (Cited on pages 2 and 51.)

[8] Francis J. Narcowich, Joseph D. Ward, Holger Wendland, *Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting*, Mathematics of Computation, 74, pp. 743-763. (Cited on page 51.)

[9] Nam Mai-Duy, Thanh Tran-Cong, *Solving high order of differential equations with radial basis function networks*, International Journal for numerical methods in Engineering, 62, pp. 824-852(2005). (Cited on pages 2, 35 and 59.)

[10] Tilmann Gneiting, *Compactly Supported Correlation Functions*, Journal of Multivariate Analysis 83, pp. 493-508 (2002). (Not cited.)

[11] P. Cardaliaguet, G. Euvrard, *Approximation of a Function and its Derivative with a Neural Network*, Neural Networks, Vol. 5, pp. 207-220(1992). (Cited on page 1.)

[12] Xin Li, *On simultaneous approximations by radial basis function neural network*, Applied Mathematics and Computation, vol. 95, pp. 75-89(1998) (Cited on pages 1, 2 and 56.)

[13] M. J. D. Powell, *Radial basis function approximations to polynomials*, Numerical analysis, (1987). (Cited on pages 57 and 67.)

[14] M. D. Buhmann, *Convergence of Univariate Quasi-Interpolation Using Multiquadrics*, IMA Journal of Numerical Analysis, Vol. 8, pp. 365-383 (1988). (Cited on pages 57 and 58.)

[15] Bengt Fornberg, Natasha Flyer, Susan Hovde, Cecile Piret, *Locality properties of radial basis function expansion coefficients for equispaced interpolation*, IMA Journal of Numerical Analysis, Vol. 28, pp. 121-142 (2008). (Cited on page 58.)

[16] J.Riley, *Solving Systems of Linear Equations with a Positive Definite, Symmetric, but possibly Ill-conditioned Matrix*, Mathematical Tables and Other Aids to Computation, 9/51, pp. 96-101(1955). (Cited on pages 2 and 65.)

[17] Gradshteyn, Ryzhik, Alan Jeffrey, Daniel Zwillinger, *Table of Integrals, Series, and Products*, Academic Press, 6th edition (August 25, 2000) (Not cited.)

[18] Charles A. Micchelli, Massimiliano Pontil, *On learning Vector-Valued Functions*, Neural Computation, Vol 17, pp.177-204(2005). (Cited on pages 57 and 67.)

[19] Rick K. Beatson, Wolfgang zu Castell, Stephan Schroedl, *Kernel based methods for vector valued data with correlated components*, Helmholtz Zentrum-Muenchen. (Cited on pages 67 and 73.)

[20] Gerold M. Grodsky, *A Threshold Distribution Hypothesis for Packet Storage of Insulin and Its Mathematical Modeling*, The Journal of Clinical Investigation, Vol. 51(1972). (Cited on page 76.)

[21] Brian Topp, Keith Promislow, Gerda De Vries, Robert M. Miura, Diane T. Finegood, *A Model of β-Cell Mass, Insulin, and Glucose Kinetics: Pathways to Diabetes*, Journal Theoretical Biology, Vol. 206, pp. 605-619(2000). (Cited on page 76.)

[22] Liene Daukste, Britta Basse, Bruce C. Baguley, David J.N. Wall, *Using a stem cell and progeny model to illustrate the relationship between cell cycle times of in vivo human tumour cell tissue populations, in vitro primary cultures and the cell lines derived from them*, Journal of Theoretical Biology, Vol. 260, pp. 563-571(2009). (Cited on page 76.)

Index

β-cell cycle, 76

Activation function, 30
Artificial neural network, 27
Artificial neural network architecture, 29
Auto-associative, 30

Bochner theorem, 13

Compartmental model, 77
Condition number, 64
Conditionally positive definite function, 19
Cover theorem, 32

Dichotomy, 32
Differential equation, 61
Direct Radial Basis Function Network (DRBFN), 38, 61
Distance Matrix, 10

Error estimates, 52
Existence theorem, 57

Feed-forward, 30

Gaussian, 10, 15, 37
Generalization phase, 28
GUI Matlab, 85

Hetero-associative, 30

Indirect Radial Basis Function Network (IRBFN), 39, 61
Interpolation problem, 9
Inverse Multiquadrics, 16, 37
Iterative method, 67

Learning phase, 31
Localization properties, 58

Matérn, 15
Matrix Conditionally positive definite, 17
Matrix kernels, 70
Multiquadrics, 20, 37, 41

Native space, 24
Neuron, 29
Non-linearly separable problem, 31

ODEs system, 73

Perceptron, 29
Positive Definite, 12
Positive Definite RBF, 12

Radial Basis Function, 10
Radial basis function network (RBFN), 34
Reproducing Kernel Hilbert Space, 22
Riley method, 67

Scaling, 66
Shannon theorem, 25
Synapse, 27, 29

Thin Plate Splines, 21

Wendland, 45

XOR example, 33